2020

Reference Governors for Time-varying Systems and Constraints

Collin Freiheit
University of Vermont

Reference Governors for Time-varying Systems and Constraints

A Thesis Presented

by

Collin J. Freiheit

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Master of Science
Specializing in Mechanical Engineering

October, 2020

Defense Date: August 25th, 2020
Thesis Examination Committee:

Hamid Ossareh, Ph.D., Advisor
Safwan Wshah, Ph.D., Chairperson
Dryver Huston, Ph.D.
Jeff Frolik, Ph.D.
Cynthia J. Forehand, Ph.D., Dean of Graduate College

Abstract

Control systems are often subject to constraints imposed by physical limitations or safety considerations, and require means of constraint management to ensure the stability and safety of the system. For real-time implementation, constraint management schemes must not carry a heavy computational burden; however many of the current solutions are computationally unattractive, especially those with robust formulations. Thus, the design of constraint management schemes with low computational loads is an important and practical problem for control engineers. Reference Governor (RG) is an efficient constraint management scheme that is attractive for real-time implementation due to its low computational complexity and ease of implementation. However, in theory, RG is only able to enforce constant constraints for systems with time-invariant models. In this thesis, we extend the capabilities of RG to solve two separate problems. The solution to the first problem presented in this thesis is a novel RG scheme for overshoot mitigation in tracking control systems. The proposed scheme, referred to as the Reference Governor with Dynamic Constraint (RG-DC), recasts the overshoot mitigation problem as a constraint management problem. The outcome of this reformulation is a dynamic Maximal Admissible Set (MAS), which varies in real-time as a function of the reference signal and the tracking output. RG-DC employs the dynamic MAS to modify the reference signal to mitigate or, if possible, prevent overshoot. The second solution presented in this thesis is a RG scheme for constraint management of parameter-varying systems with slowly time-varying constraints. The solution, known as the Adaptive-Contractive Reference Governor (RG-AC) utilizes a contractive characterization of MAS that changes in real-time as a function of the system's time-varying parameters in a computationally attractive manner. This adaptive set is based off a first-order Taylor approximation of the parameter dependent matrices that describe the time-varying MAS. The work in this thesis is supported by simulation results which demonstrate the efficacy of both approaches, and also highlight their limitations.

For my family and my friends. Thank you for your support.

Acknowledgements

I would like to thank my advisor, Dr. Hamid Ossareh for his guidance, attention, and suggestions. I would also like to thank Aidan, Joycer, and Yudan for their willingness to field questions and collaborate on challenging problems. Finally, I would like to thank my committee, Dr. Ossareh, Dr. Wshah, Dr. Huston, and Dr. Frolik, for playing an important role towards my degree and making the effort to learn about my research.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Constraints are often present in practical control systems and can cause performance degradation of systems or even disastrous consequences if overlooked. Consider a radio-controlled multi-rotor drone for example. In the presence of a rapid reference change, the voltage command to one or more of the drone's motors could saturate and cause poor performance or instability, resulting in a crash. Constraints are often identified or established on actuator components, similar to the drone's motors in the previous example, or system states, for example, the drone's pitch and roll angles. Some constraints may be handled by over-designing the hardware of a system, however this can be expensive or impractical, and does not necessarily guarantee constraint satisfaction. Thus alternative methodologies are desired for constraint management of control systems. Over the past couple of decades, theoretical and practical contributions have been made towards methods for constraint enforcement of control systems. Some of these methods include Barrier Lyapunov Functions (BLF) [1–3], Model Predictive Control (MPC) [4–10], and Reference Governors (RG) [11–21]. To provide a brief background, Barrier Lyapunov Functions are a modification to Lyapunov based control where bounds are set on the arguments of BLF such that as a state approaches its boundary, BLF approaches infinity. MPC is a control methodology that uses predictions of a system's state trajectories to optimize a control input to stabilize a system while satisfying a set of constraints. Similarly, RG is a predictive control strategy that employs a prediction of the evolution of the system state to enforce pre-specified constraints on the inputs, states, or the outputs. Unlike MPC, however, RG modifies the reference to a pre-stabilized closed-loop control system and

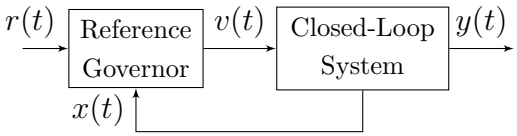

Figure 1.1: Reference governor block diagram.

is primarily intended for constraint management. Moreover, RG is more numerically attractive than MPC and unlike BLF, does not require the existence of a Lyapunov function, which makes it attractive for real-time control, especially that of fast processes. A block diagram of a closed-loop system controlled by RG is depicted in Fig. 1.1. To briefly summarize, RG employs the so-called Maximal Admissible Set (MAS) [22], which characterizes the set of all initial conditions and constant inputs that satisfy the constraints for all time. The MAS is computed offline, allowing the RG to enforce the constraints in real-time by solving an explicit linear program subject to state and input values belonging to the MAS. An in-depth review of RG and MAS can be found in Section 2.1.1.

A limiting factor of RG, however, is that it can only enforce constant constraints for systems with time-invariant models. This poses a problem because, in practice, systems may have time-varying models, and constraints may be time-varying. For example, the dynamical model of an airplane is, in general, time-varying because the mass of the system changes due to the discharge of burned fuel from the engines. Furthermore, at different altitudes and different speeds, constraints on control surface angles or attitude may change for safety reasons. Thus, due to the lack of current literature for constraint management for systems with time-varying constraints and time-varying dynamics, the computational simplicity of RG, and the practical need for such constraint management schemes, this thesis focuses on the development of RG for

time-varying systems and/or time-varying constraints. Specifically, we propose two problems related to constraint management for systems with time-varying dynamics and/or time varying constraints, and solve them with adaptations of RG that employ time-varying characterizations of MAS. The first problem deals with overshoot mitigation for tracking control systems. In our solution, we develop the so-called Reference Governor with Dynamic Constraint (RG-DC), which recasts the overshoot mitigation problem as a constraint management problem with a time-varying constraint. The outcome of this reformulation is a dynamic Maximal Admissible Set (MAS), which varies in real-time as a function of the reference signal and the tracking output. RG-DC employs the dynamic MAS to modify the reference signal to mitigate or, if possible, prevent overshoot. The second problem deals with constraint management for systems with slowly time-varying constraints and slowly time-varying dynamics; where the time-varying nature of the system is characterized by time-varying parameters in the linear system model. The solution known as the Adaptive-Contractive Reference Governor (RG-AC) utilizes a contractive characterization of MAS that changes in real-time as a function of the system's time-varying parameters in a computationally attractive manner. This adaptive set is based off a first-order Taylor series approximation of the parameter dependent matrices that describe the time-varying MAS. Note that [23] first proposed an RG scheme for slowly-time varying constraints, however we extend this work to also handle slowly parameter-varying systems. Overall, this thesis makes contributions to the literature of control theory, specifically constraint management, set theoretic methods, and reference governors.

1.2 Problem Statements

As stated above, this thesis is concerned with two problems related to constraint management for time-varying systems and/or time-varying constraints. In both solutions, adaptations of RG are developed such that real-time implementation is viable. In this section we further motivate and propose both problem statements.

1.2.1 Problem Statement 1: Overshoot Mitigation for Tracking Systems

Overshoot in closed-loop control systems is often an undesired phenomenon. For example, position overshoot in servo controlled robots may result in collisions, and in regulated electronic power converters, overshoot may cause overload currents. Surprisingly, there are very few methods available in the literature of control systems dedicated to overshoot mitigation. One obvious solution is feedforward plant inversion [24, 25], wherein a pre-filter is used to eliminate the overshoot resulting from the underdamped and/or zero dynamics of the closed-loop system. However this strategy requires an exact model of the plant, which is not always available. Additionally, a stable non-minimum phase system poses the problem of system destabilization upon plant inversion. Another strategy is to use a detuned or a more complex controller within the loop; however, this approach has the downside of slowing down the system, increasing its complexity, or not being able to handle variability in the plant dynamics. Furthermore, this approach may not be applicable to off-the-shelf products or systems with legacy controllers. Other overshoot mitigation solutions in the literature

include a cascade control scheme coupled with a sliding mode controller [26], and a feedback gain design method based on quantifier elimination [27]. These solutions either require an accurate model of the plant or increase the complexity of the inner loop. In this thesis, we propose a novel overshoot mitigation strategy using the Reference Governor (RG) framework. Unlike the existing methods in the literature, the proposed strategy does not require modifications to the controller within the closed-loop system, does not require model inversion, and can be made robust to modeling errors.

Traditional RG theory can only handle static constraints (i.e., constraints that do not vary with time). In this thesis, we cast overshoot as a dynamic (i.e., time-varying) constraint on the tracking output (denoted by y_{tr} hereafter) of the system. Specifically, if $y_{tr}(t)$ is above $r(t)$, we wish to maintain y_{tr} above $r(t)$ for all future time. Similarly, if $y_{tr}(t)$ is below $r(t)$, we wish to hold y_{tr} below $r(t)$ for all future time. To accomplish this in the framework of the reference governor, we define the constraint set $\{y : y \leq r(t)\}$ whenever $y_{tr}(t) \leq r(t)$, and by the set $\{y : y \geq r(t)\}$ whenever $y_{tr}(t) > r(t)$. This dynamically-varying constraint leads to a novel, dynamically-varying MAS. We present a unique modification of the RG theory to allow it to handle such dynamic MAS. We call this RG solution the Reference Governor with Dynamic Constraint (RG-DC).

The dynamic nature of the MAS and our RG-DC formulation raise the following questions:

1. Does the number of inequalities required to describe the dynamic MAS change as the reference varies in real-time?

2. What is the geometric and algebraic relationship between the instances of the

dynamic MAS at different times?

3. Does RG-DC guarantee constraint satisfaction for all time?

4. Can RG-DC destabilize the control loop?

5. How much additional computational complexity does RG-DC introduce compared to standard RG?

6. How can RG-DC be made robust to model uncertainty and unknown disturbances?

All of these questions will be addressed in Chapter 3.

Note that reference [23] investigates a RG solution for systems with slowly-varying constraints. However, the results of [23] are not applicable to our problem because the dynamic constraint considered in this thesis may vary rapidly and is designed to mitigate overshoot. Furthermore, similar to this thesis, reference [28] briefly considers overshoot mitigation in the framework of RG, but it does not provide a rigorous answer to the questions raised above.

1.2.2 Problem Statement 2: Constraint Management for Slowly Time-varying Systems and Constraints

Time-varying systems pose a problem for constraint management techniques. All schemes mentioned in Section 1.1, namely BLF, MPC, and RG use some sort of mathematical representation to describe the behavior of the system. Embedded in

these equations, are parameters associated to the physical system. If these parameters change over time, then the mathematical models used to design the constraint management schemes may not be accurate during later operation. This could result in poor constraint management performance and even constraint violation. Consider the attitude control of a rocket for example. As the rocket expels fuel, the mass and moments of inertia of the rocket change. Thus, the same attitude adjust input to the rocket will result in different outputs when the rocket is full of fuel at the beginning of its ascent vs. when it is low on fuel during a later stage of launch. Therefore if a constraint management scheme is designed for a rocket with a full fuel tank, a constraint admissible input at the beginning of the launch may not be admissible later.

Only a few solutions for constraint management for time-varying systems have been proposed in the literature. In [29], an algorithm to construct maximal robust positively invariant sets for linear systems with polytopic model uncertainty is proposed. In this paper, model uncertainty is characterized by a linear time-varying autonomous system where the time-varying A matrix belongs to a known uncertainty polytope. In [30], an algorithm to calculate admissible sets for dynamical systems with a single slowly-varying parameter is proposed. In both solutions ([29] and [30]), the admissible sets are calculated offline and do not change as the system operates. Furthermore the robust invariant sets formed by each method are subsets of the maximal admissible set of the system if it were updated every timestep based on the time-varying parameters. Therefore, the governed responses using the set formulations in [29] and [30] may be conservative.

Ideally, to achieve the best constraint admissible response, MAS would be recal-

culated every timestep; however this is impractical for the control of fast processes because the computation of MAS is complex and highly system dependent. Thus, to increase constraint management performance and avoid a conservative response we propose a characterization of MAS that changes in real-time as a function of the time-varying parameters without recalculation of MAS at every timestep. This time-varying characterization of MAS is hereafter referred to as Parameter-Dependent MAS. To generate Parameter-Dependent MAS, a first-order Taylor approximation of the parameter-dependent matrix terms that describe the time-varying MAS is calculated at the nominal parameter values. Thus if we are able to measure or estimate the values of the time-varying parameters at every timestep, we can update Parameter-Dependent MAS to approximate the actual time-varying MAS in a computationally friendly manner. Note that Parameter-Dependent MAS is not necessarily maximal or admissible, but we proceed with the nomenclature for simplicity.

A time-varying admissible set poses the problem of constraint violation if the admissible set shrinks in any manner. In [23], contractive constraint admissible sets are proposed for systems with slowly time-varying constraints such that if the constraint set shrinks within predetermined rate limits, constraint satisfaction is guaranteed for all time. We thus adopt the contractive RG methods proposed in [23], along with our formulation of Adaptive MAS to form Adaptive-Contractive Reference Governor (RG-AC).

The dynamic nature of Adaptive-Contractive MAS and our RG-AC formulation raise the following questions:

1. Does the number of inequalities required to describe Parameter-Dependent MAS change as the parameters vary in real-time?

2. Does RG-AC guarantee constraint satisfaction for all time?

3. How much additional computational complexity does RG-AC introduce compared to standard RG?

4. Can RG-AC destabilize the control loop?

All of these questions will be addressed in Chapter 4.

Note that [31] Considers a RG scheme for parameter-varying systems called the Parameter-Adaptive Reference Governor (PARG) that, similar to RG-AC, utilizes a time varying characterization of MAS. The differences between RG-AC and PARG however are listed below:

- PARG is designed for nonlinear systems whereas RG-AC is designed for linear systems

- PARG relies on offline simulation to generate potentially large data sets whereas RG-AC explicitly generates only a few matrices that describe the time-varying characterization of MAS

- PARG utilizes a support vector machine algorithm along with a quadratic program to calculate an admissible reference whereas RG-AC exploits an explicit linear program

Finally, note that reference [32] considers a RG scheme for the specific purpose of fuel cell starvation protection, in which they assume parametric uncertainties in the system model and use an approach based on sensitivity functions to model the systems behaviour. However the sensitivity functions in [32] are used to describe the changes in the nominal output trajectory as a function of the parameters, unlike in our

formulation where the sensitivity functions are used to characterize the changes in the edges of the nominal MAS. In fact, [32] does not employ an admissible set, instead they use a bisectional search algorithm to iteratively bisect the decision variable, simulate the system based on the linear approximation, and check for predicted constraint violations. As a result the RG scheme proposed in [32] can be very conservative.

1.3 Contributions

The original contributions of this thesis are to the literature of control theory, specifically constraint management, set theoretic methods, and reference governors. The contributions are separated by chapter. Problem Statement 1 is addressed in Chapter 3, and Problem Statement 2 is addressed in Chapter 4. We explain the contributions below.

The main contributions of Chapter 3 are a new approach (RG-DC) to mitigate overshoot in closed-loop control systems, and the analysis and demonstration of the six questions raised in Section 1.2.1. Additionally, we reveal an interesting property of RG-DC regarding its effect on the governed system's frequency response. Specifically, RG-DC can act as a novel nonlinear filter to eliminate resonance in closed-loop systems caused by underdamped poles and/or zero dynamics.

The main contributions of Chapter 4 are a new approach (RG-AC) to enforce slowly time-varying constraints on slowly parameter-varying systems, and the analysis of the four questions raised in Section 1.2.2.

1.4 Thesis Outline

Chapter 2 contains a literature review where we provide an in-depth review of maximal admissible sets and reference governors; specifically scalar reference governor (RG), vector reference governor (VRG), and Command Governor (CG). Furthermore, we review admissible sets for time-varying systems and admissible sets for time-varying constraints. The main results of this thesis are covered in Chapters 3 and 4 where we formally present RG-DC and RG-AC respectively. Finally, in Chapter 5, we summarize the thesis and propose ideas for future research.

Chapter 2

Preliminaries

2.1 Literature Review

In this section, we present a review of the current state of the literature of reference management schemes for linear systems. Specifically, we review maximal admissible sets, reference governors, vector reference governors, and command governors. Furthermore we review some of the current solutions in the literature for admissible sets for time-varying systems and admissible sets for time-varying constraints.

2.1.1 Reference Management Schemes

Reference Governor was first proposed by Kapasouries in 1988 [33]. Since then, several reference management techniques have been proposed and adopted by the community for both linear and nonlinear systems. Some of these techniques for linear systems include, but are not limited to: Scalar Reference Governor, Vector Reference Governor, and Command Governor. In this section, we explore the formulations of all three reference governor schemes as well as the foundation to each scheme known as the Maximal Admissible Set (MAS). It should be noted that Reference Governor was first proposed in the continuous-time framework, however a natural extension to the discrete-time domain has been widely adopted due to its mathematical simplicity and capability for real-time implementation. Thus, in this thesis we exclusively focus on the discrete-time formulations for reference governors.

Maximal Admissible Set (MAS)

The MAS of a system is defined as the set of all initial states and constant inputs, such that the system's output constraints are satisfied for all future time. In this section we derive the formulation of MAS and present an algorithm to compute it.

Consider Fig. 1.1, and let the "closed-loop system", in general, be described by the multi-input multi-output discrete-time, stable linear system:

$$x(t+1) = Ax(t) + Bv(t)$$
$$y(t) = Cx(t) + Dv(t)$$

where the output $y(t)$ is subject to the following polyhedral constraints:

$$y(t) \in \mathbb{Y} \triangleq \{y : Sy \leq s\} \tag{2.1}$$

Note that vector inequalities here and throughout the thesis are to be interpreted element-wise. In general, the set in (2.1) may be unbounded. RG employs the so-called maximal admissible set (MAS), denoted by O_∞, which, to reiterate, is the set of all initial states and constant control inputs that satisfy (2.1) for all time:

$$O_\infty = \left\{(x, v) : x(0) = x,\ v(t) = v,\ y(t) \in \mathbb{Y},\ \forall t \in \mathbb{Z}^+\right\} \tag{2.2}$$

As seen in (2.2), to construct MAS, $v(t) = v$ is held constant for all t. Using this assumption, the evolution of the output $y(t)$ can be expressed explicitly as a function

of $x(0) = x$ and v:

$$y(t) = CA^t x + \Big(C(I - A^t)(I - A)^{-1}B + D\Big)\, v \tag{2.3}$$

Therefore, MAS in (2.2) can be characterized by a polyhedron defined by an infinite number of inequalities:

$$O_\infty = \Big\{(x, v) : SCA^t x + S\left(C(I - A^t)(I - A)^{-1}B + D\right) v \leq s,\ \forall t \in \mathbb{Z}^+\Big\} \tag{2.4}$$

It is shown in [22] that, under mild assumptions on C and A, it is possible to make this set finitely determined (i.e., be described by a finite number of inequalities) by constraining the steady-state value of y, denoted by $y(\infty)$, to the interior of the constraint set:

$$y(\infty) \triangleq \Big(C(I - A)^{-1}B + D\Big)\, v \in (1 - \epsilon)\mathbb{Y} \tag{2.5}$$

where $\epsilon \in \mathbb{R}^+$ is a small number. As shown in [22], after introducing (2.5) in MAS, there exists a finite prediction time j^*, where the inequalities corresponding to all future prediction times ($t > j^*$) are redundant. The smallest such j^* is referred to as the *admissibility index* of the MAS.

Combining (2.4) and (2.5), we obtain an inner approximation of O_∞, denoted by $\widetilde{O}_\infty$, which can be represented by:

$$\widetilde{O}_\infty = \Big\{(x, v) : H_x x + H_v v \leq h\Big\} \tag{2.6}$$

where the matrices H_x, H_v, and h are finite dimensional, and have the following form:

$$H_x = \begin{bmatrix} 0 \\ SC \\ SCA \\ SCA^2 \\ \vdots \\ SCA^{j^*} \end{bmatrix}, \quad H_v = \begin{bmatrix} S\left(C(I-A)^{-1}B + D\right) \\ SD \\ S\left(C(I-A)(I-A)^{-1}B + D\right) \\ S\left(C(I-A^2)(I-A)^{-1}B + D\right) \\ \vdots \\ S\left(C(I-A^{j^*})(I-A)^{-1}B + D\right) \end{bmatrix}, \quad h = \begin{bmatrix} (1-\epsilon)s \\ s \\ s \\ s \\ \vdots \\ s \end{bmatrix} \tag{2.7}$$

To numerically construct H_x, H_v and h, we begin with the steady-state inequality in (2.5) and iteratively add the inequalities in (2.4) starting with $t = 0$. After each t, we check if the newly added rows are all redundant. If this is so, j^* has been reached and the construction of $\widetilde{O}_\infty$ is complete.

We now review the algorithm provided in [22] to check for redundancy. Note that this algorithm is leveraged in Sections 3.1 and 4.1 for the analysis of our dynamic characterizations of MAS. Given any polyhedron defined by $Mz \leq N$ and a scalar inequality given by $c^T z \leq d$, to determine if the inequality is redundant with respect to the polyhedron, it is common practice to solve the following linear program (LP) [22]:

$$f = \max \; c^T z \quad \text{subject to} \quad Mz \leq N \tag{2.8}$$

If $f \leq d$, the new inequality is redundant. To apply this idea to MAS, suppose MAS has been partially constructed with the inequalities in (2.5) and (2.4) from $t = 0$ up to $t = j$, for some j. Let H_x, H_v, h represent the matrices of this partially constructed MAS. We wish to test whether an inequality in (2.4) with $t = j+1$ is redundant. The

LP above can be used for this purpose, with $M = [H_x, H_v]$, $N = h$, $z = (x, v)$, and c^T and d representing the inequality being tested for redundancy. For completeness, a detailed algorithm to calculate $\widetilde{O}_\infty$ is presented in Algorithm 1.

Algorithm 1 $\widetilde{O}_\infty$

Inputs:
$A, B, C, D, S, s, \epsilon$
Outputs:
H_x, H_v, h, j^*

initialize H_x as 0
initialize H_v as $S\left(C(I-A)^{-1}B + D\right)$
initialize h as $(1-\epsilon)s$
initialize j as 0
initialize flag as 0
while flag = 0 **do**
  $H_x^* := SCA^j$
  $H_v^* := S\left(C(I-A^j)(I-A)^{-1}B + D\right)$
  $h^* := s$
  initialize i as 1;
  for each row in h^* **do**
    $f^*(\text{row } i) := \underset{(x,v)}{\text{maximize}} \quad H_x^*(\text{row } i)x + H_v^*(\text{row } i)v$
    $\text{s.t.} \qquad H_x x + H_v v \leq h$
    $i := i + 1$
  end for
  if each row of f^* corresponds to a bounded solution and $f^* \leq h^*$ **then**
    $j^* := j - 1$
    set flag as 1
  else
    vertically concatenate H_x^* to the bottom of H_x, this becomes the new H_x matrix
    vertically concatenate H_v^* to the bottom of H_v, this becomes the new H_v matrix
    vertically concatenate h^* to the bottom of h, this becomes the new h matrix
    $j := j + 1$
  end if
end while

Scalar Reference Governor (RG)

RG, first introduced by Gilbert in 1994 [34], is designed for single-input multi-output, discrete-time, stable linear systems of the form:

$$x(t+1) = Ax(t) + Bv(t)$$
$$y(t) = Cx(t) + Dv(t)$$

where the constraints may exist on the outputs. In [34], the command $v(t)$ (see Fig. 3.1.2) is governed by:

$$v(t) = \kappa r(t) \tag{2.9}$$

where the parameter $\kappa \in [0, 1]$ is optimized such that the states and modified reference command belong to MAS, that is $(x(t), v(t)) \in \widetilde{O}_\infty$, while minimizing the distance between $v(t)$ and $r(t)$. This RG scheme, known as static RG, has the disadvantage of generating oscillations on the command signal $v(t)$ during threats of constraint violation. Improvements were made to static RG starting in 1995 through the works of Bemporad [35], and later through Gilbert and Kolmanovsky, [12] and [14], in 1995 and 1999 respectively. Based on (2.3), in the formulation for $\widetilde{O}_\infty$, the future predictions of the state trajectories are based off a constant command $v(t) = v$. Therefore, if $v(t-1)$ was an admissible reference command at timestep $t-1$, then according to MAS, $v(t) = v(t-1)$ is also an admissible reference command at timestep t. Thus, and adaptation to the static RG update law (2.9) was made such that $v(t)$ became the convex combination of $r(t)$ and $v(t-1)$ instead of $r(t)$ and 0. This modification eliminated the oscillatory behavior of the reference command $v(t)$ and was adopted

as the RG update law, replacing (2.9). The RG update law is presented below.

$$v(t) = v(t-1) + \kappa\left(r(t) - v(t-1)\right) \tag{2.10}$$

where again, $\kappa \in [0, 1]$. To select κ, we solve the following linear program in an effort to drive $v(t)$ as close to $r(t)$ as possible, without violating constraints:

$$\begin{aligned} \underset{\kappa\in[0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & v(t) = v(t-1) + \kappa\left(r(t) - v(t-1)\right) \\ & (x(t),\ v(t)) \in \widetilde{O}_\infty \end{aligned} \tag{2.11}$$

where $x(t)$, $r(t)$, and $v(t-1)$ are known parameters at time t. Note that (2.11) has an explicit solution due to the structure of the optimization problem's constraints, which is the main factor behind the numerical appeal of RG. Furthermore, note that if $\kappa = 0$, the control command from the previous timestep is maintained to avoid constraint violation, and if $\kappa = 1$, the reference $r(t)$ is feasible and, therefore, $v(t) = r(t)$. For completeness, the algorithm executed online by RG at every timestep is presented in Algorithm 2.

Vector Reference Governor (VRG)

Note that for a system with multiple inputs, RG can be implemented, however since (2.11) solves for one optimization variable, namely κ, the solution will prioritize the input with the highest threat of constraint violation and therefore lower the input values of all channels by the same factor. This is likely to cause an overly conservative response because some inputs could be over-governed. The solution to reduce

Algorithm 2 RG

Inputs:
$r(t), x(t), v(t-1), H_x, H_v, h$
Output:
$v(t)$

initialize j as 1
initialize κ as a vector with the same number of rows as h
for each row in $\widetilde{0}_\infty$ **do**
    $n := h(\text{row } j) - H_x(\text{row } j)x(t) - H_v(\text{row } j)v(t-1)$
    $d := H_v(\text{row } j)(r(t) - v(t-1))$
    $\kappa(\text{row } j) := \text{kappa}(n, d)$
    $j := j + 1$
end for
$\kappa^* := \min(\kappa)$
$v(t) := v(t-1) + \kappa^*(r(t) - v(t-1))$

function kappa(n, d)

if $n > 0$ **then**
    if $d > 0$ **then**
        $\kappa := \min(n/d, 1)$
    else
        $\kappa := 1$
    end if
else
    $\kappa := 0$
end if
return κ

end function

the conservatives of RG for systems with multiple inputs lies within the option to individually optimize each input channel. This is the formulation behind VRG [36]. The VRG uses a diagonal matrix $\boldsymbol{K} \in \mathbb{R}^{m \times m}$ (where m is the dimension of v) to command each input channel independently. The VRG command update law is defined as:

$$v(t) = v(t-1) + \boldsymbol{K}\left(r(t) - v(t-1)\right) \tag{2.12}$$

where $\boldsymbol{K} = \text{diag}(\kappa_i(t))$. Note that the values $\kappa_i(t)$ for all $i \in \{1, \ldots, m\}$ are optimized by solving the following quadratic program:

$$\begin{aligned} \underset{\kappa_i \in [0,1]}{\text{minimize}} \quad & \|v(t) - r(t)\|_Q \\ \text{s.t.} \quad & v(t) = v(t-1) + \boldsymbol{K}\left(r(t) - v(t-1)\right) \\ & (x(t),\ v(t)) \in \widetilde{O}_\infty \end{aligned} \tag{2.13}$$

where $Q = Q^T > 0$.

Command Governor (CG)

Command Governor [37,38] tackles the same problem as VRG, however CG computes $v(t)$ directly via the following optimization problem:

$$\begin{aligned} \underset{v}{\text{minimize}} \quad & \|v(t) - r(t)\|_Q^2 \\ \text{s.t.} \quad & (x(t),\ v(t)) \in \widetilde{O}_\infty \end{aligned} \tag{2.14}$$

Where $Q > 0$. Because CG directly optimizes $v(t)$, it can provide a faster response than RG if $m \geq 2$. However the improved performance of CG compared to RG comes at the price of increased computational complexity of the optimization problem.

2.1.2 Applications of Reference Governors

Because reference governors have the capability of theoretically guaranteeing the enforcement of constraints with little computational effort and no modification to the closed loop system's controller, they have become relatively successful in applications involving fast dynamics and platforms with limited computational ability. As a re-

sult RG has been adopted in areas including the automotive and aerospace industries as well power grids. Note that the following examples are only some of the documented applications of reference governors. For a more comprehensive review of the applications of reference governors see [13].

Automotive Applications

Due to increasingly stringent regulations, and limited computational bandwidth allotted to different devices within vehicles, reference governors have found numerous applications in the automotive industry.

One of the most prevalent recorded uses for reference governors in the automotive industry has been to address compressor surge constraints in turbocharged gasoline engines, which is the subject of references [28], [39], and [40]. Other effective applications for reference governors in vehicles include constraint handling in diesel engines [41, 42] and, for non-conventional powertrains, preventing oxygen starvation in fuel cells [32, 43–45], and handling constraints in electric batteries [46, 47]. Finally, reference [48] addresses vehicle rollover prevention.

Aerospace Applications

Actuator saturation in aerospace applications can be detrimental to flight stability. As a result, reference governors schemes have been applied to numerous areas within the aerospace industry to mitigate actuator limit and rate saturation (see references [49–51] and [52]). Additionally, reference governors have been applied to quad-rotors for obstacle avoidance in references [53] and [54].

Power Grid Applications

In recent years reference governor strategies have been adopted for use in power networks for load frequency regulation and voltage regulation. The former has been addressed in [55, 56] and [57], whereas the later has been addressed in [58] and [59]. With the growing popularity of Distributed Energy Resources (DERs) and increasingly stringent grid regulations, the popularity of constraint management schemes for power grid applications is likely to increase.

2.1.3 Admissible sets for Time-Varying Systems

Only a few schemes have been proposed in the literature to develop admissible sets for time-varying systems. In [29], an efficient algorithm to construct maximal robust positively invariant sets for linear systems with polytopic model uncertainty is proposed. In this paper, model uncertainty is characterized by a linear time-varying autonomous system where the time varying A matrix belongs to a known uncertainty polytope. The resulting set is shown to consist of a finite number of halfspaces if the system model is quadratically stable within the bounds of model uncertainty. The resulting sets are also shown to be larger than ellipsoidal invariant sets formed from equivalent system models.

In [30], an algorithm to calculate admissible sets for dynamical systems with slowly-varying parameters is proposed. More specifically, this paper exploits the slowly-varying nature of the system to develop the notion of backward-reachable sets, which are used in the algorithm to generate maximal admissible sets for time-varying autonomous systems. It is stated in the paper that the maximal admissible sets

produced by the algorithm are supersets of robust positively invariant sets, which do not consider bounds on the time-varying rates of system parameters. Furthermore, it should be noted that the time-varying parameters don't need to be observable to form the admissible set. In both [29] and [30], it is stated that developing admissible sets for higher order time-varying systems becomes computationally demanding.

Reference [31] Considers an RG scheme for parameter-varying nonlinear systems called the Parameter-Adaptive Reference Governor (PARG) that, unlike [29] and [30], employs an admissible set that changes in real-time. PARG utilizes a support vector machine algorithm (see [60–62] for reference) that dynamically learns constraint admissible sets by combining off-line simulation data based on sampling, and online data provided by a parameter estimator based on Bayesian update laws (see [63] for reference). It should be noted that the offline simulation data generated for use by PARG online has the potential to grow large depending on the order of the closed-loop system model and the resolution of the grid used for sampling. This could pose a problem if memory is limited on the hardware used for implementation. In [31], the efficacy of PARG was demonstrated on a second-order nonlinear system and it was shown in simulation that PARG was able to satisfy constraints by employing an adaptive admissible set whereas a non-adaptive RG violated constraints.

2.1.4 Admissible sets for Time-Varying Constraints

In [64] a real-time MPC-based tracking strategy for linear systems subject to time-varying constraints is proposed. To handle the problem, a polytopic invariant set computed offline is homogeneously dilated and contracted on-line to fit the polytopic time-varying constraints, and used as an admissible terminal set constraint to guarantee stability and convergence in the tracking task. It is mentioned that the on-line cost of the homothetic invariant set computation is negligible compared to the computational demand of MPC. In [39], a RG based approach for enforcing slowly time-varying constraints is proposed. The RG formulation is based on the so called λ-contractive set, which allows the constraint admissible set to decay quickly enough to enforce slowly time-varying constrains. The λ-contractive constraint admissible set is shown to be a subset of the maximal admissible set. A detailed review of λ-contractive admissible sets appears later in Chapter 4, as they play an important role in the formulation of one of the main results of this thesis.

Chapter 3

RG-DC: Overshoot Mitigation Reference Governor

3.1 Reference Governor with Dynamic Constraint (RG-DC)

We now present the first contribution of this thesis, namely the RG-DC, which mitigates overshoot of tracking control systems by casting overshoot as a constraint management problem and employing a dynamic MAS which changes in real-time as the constraints on the tracking output change.

Consider the asymptotically stable system

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t) \\ y_{tr}(t) &= C_{tr}x(t) + D_{tr}v(t) \\ y_{st}(t) &= C_{st}x(t) + D_{st}v(t) \end{aligned} \tag{3.1}$$

with DC gain from v to y_{tr} equal to 1, where $y_{tr} \in \mathbb{R}$ is the tracking output on which we wish to enforce the dynamic overshoot constraint (as explained below). Additionally, $y_{st} \in \mathbb{R}^p$ refers to constrained outputs, on which we wish to enforce standard static constraints:

$$y_{st}(t) \in \mathbb{Y}_{st} \triangleq \{y : S_{st}y \leq s_{st}\} \tag{3.2}$$

It should be noted that, because y_{tr} is the output of the plant within the closed-loop system, there is no feedforward from v to y_{tr} in practice. Thus, for the remainder of the chapter, we assume that $D_{tr} = 0$. Note however that D_{st} is allowed to be non-zero because static constraints could be imposed on controller states or the controller

output, which may require feedthrough.

For overshoot mitigation, we impose that y_{tr} be constrained by the reference r, which may vary with time. To do so, two cases must be considered: the first case is where $y_{tr}(t) \leq r(t)$ at the current time t, for which we define overshoot by the following condition: $\exists k > t$ such that $y_{tr}(k) > r(t)$. Thus, to prevent overshoot, we must enforce the following constraint: $y_{tr}(k) \in \{y : y \leq r(t)\}$ for all $k > t$. In the second case, $y_{tr}(t) > r(t)$ at the current time t, for which we define overshoot by $\exists k > t$ such that $y_{tr}(k) < r(t)$ and the constraint by $y_{tr}(k) \in \{y : y \geq r(t)\}$ for all $k > t$. Note that we have chosen the constraint sets to be closed (i.e., the inequalities are not strict), which is necessary to ensure that the linear programs that arise in RG-DC are well-posed. The above leads to a time-varying constraint set that depends on both $y_{tr}(t)$ and $r(t)$:

$$\mathbb{Y}_{tr}(r(t),\ y_{tr}(t)) \triangleq \begin{cases} \{y : y \leq r(t)\} & y_{tr}(t) \leq r(t) \\ \{y : y \geq r(t)\} & y_{tr}(t) > r(t) \end{cases} \tag{3.3}$$

The goal is to enforce $y_{tr}(k) \in \mathbb{Y}_{tr}(r(t),\ y_{tr}(t))$ for all $k > t$.

We now define the maximal admissible sets for this system. For the static constraint in (3.2), we create MAS as discussed previously in Section 2.1.1. We denote this MAS by $O_{\infty,st}$. For the dynamic MAS, note that the second constraint in (3.3) can be re-written as $\{y : -y \leq -r(t)\}$, which implies that both constraints in (3.3) can be cast in the form (2.1), where S takes on the values of 1 or -1 and s takes on the values of $r(t)$ or $-r(t)$. Therefore, the definition of MAS remains the same as (2.2), with the exception that, since $\mathbb{Y}_{tr}$ depends on $r(t)$ and $y_{tr}(t)$, so does the MAS. We thus denote this dynamic MAS by $O_{\infty,tr}\left(r(t), y_{tr}(t)\right)$. In Subsection 3.1.1,

we analyze the properties and computation of this dynamic MAS.

The proposed reference governor scheme (RG-DC) employs the intersection of the static MAS (for constraints on y_{st}) and the dynamic MAS (for constraints on y_{tr}) to compute κ from (2.11) and $v(t)$ from (2.10). We will discuss the stability and recursive feasibility of the system with RG-DC, as well as the implementation aspects, in Subsection 3.1.2. We also discuss a robust formulation of RG-DC to handle plant-model mismatch and unknown disturbances.

For simplicity, we assume that all states of the system are available for feedback. If not, a set-based observer can be designed as is done in [39].

3.1.1 Computational aspects and properties of the dynamic MAS

We first address the computation of the dynamic MAS defined above (the computation of the static MAS is standard and will not be addressed). For this investigation, we seek to develop a polyhedral characterization of the dynamic MAS, parameterized on $r(t) = r$ and $y_{tr}(t)$.

First suppose that $r > 0$ denoted r^+. We will relax this assumption later. Now consider the inequalities in (2.4). Recall from above that S takes on the value of 1 (in which case $s = r^+$), or -1 (in which case $s = -r^+$). If $S = 1$, the steady-state halfspace should be shrunk to: $v \leq (1 - \epsilon)r^+$ and the inequalities in (2.4) become:

$$C_{tr}A^t x + C_{tr}(I - A^t)(I - A)^{-1}Bv \leq r^+ \tag{3.4}$$

If $S = -1$, the steady-state halfspace should be shrunk to: $v \geq (1 + \epsilon)r^+$ and the

inequalities in (2.4) become:

$$C_{tr}A^t x + C_{tr}(I - A^t)(I - A)^{-1}Bv \geq r^+ \tag{3.5}$$

A polyhedral representation of MAS constructed from the tightened steady-state constraint $v \leq (1 - \epsilon)r^+$ and the inequalities in (3.4) for all $t \geq 0$ is given by:

$$O_\infty^-(r^+) = \left\{(x, v) : H_x x + H_v v \leq r^+ h^-\right\} \tag{3.6}$$

Similarly, a representation of MAS using (3.5) with the tightened steady-state constraint $v \geq (1 + \epsilon)r^+$ is:

$$O_\infty^+(r^+) = \left\{(x, v) : H_x x + H_v v \geq r^+ h^+\right\} \tag{3.7}$$

where h^- and h^+ are vectors of all 1s except the first block of rows, which are $1-\epsilon$ and $1 + \epsilon$, respectively. Note that in order to explicitly show the dependence of the sets on r^+, we have formulated (3.6)-(3.7) with r^+h^- and r^+h^+ on the right hand sides (instead of simply h as in (2.6)). For now, we consider H_x, H_v, h^-, h^+ as being infinite dimensional matrices (i.e., a redundancy check was not performed when forming O_∞^- and O_∞^+). Since we know, from Section 2.1.1, that both (3.6) and (3.7) must be finitely determined for a fixed r^+, our goal now is to study the admissibility index of these sets as functions of r^+.

Recall from Section 2.1.1 that to find the admissibility index of a MAS, we construct it row by row and stop when redundancy is detected. Furthermore, to detect redundancy, we use the linear program (LP) in (2.8). While redundancy can be

checked for $O_\infty^-(r^+)$ using the same approach, $O_\infty^+(r^+)$ requires a LP of a different form. To formulate a LP for $O_\infty^+(r^+)$, we represent (3.7) in the form of (3.6), yielding $-(H_x x + H_v v) \leq -r^+ h^+$. Upon applying (2.8) to this inequality and simplifying the resulting LP, we obtain the following adaptation of (2.8):

$$f = \min \; c^T z \quad \text{subject to} \quad Mz \geq N \tag{3.8}$$

To proceed with our analysis of admissibility index, we first show, with support of Lemma 1, that the individual admissibility indices of $O_\infty^-(r^+)$ and $O_\infty^+(r^+)$ are unchanged for any r^+.

Lemma 1. *Suppose the unique maximizer of*

$$\max \; c^T z \quad \text{subject to} \quad Mz \leq N \tag{3.9}$$

is given by z^*. Then, for any $\gamma \in \mathbb{R}^+$, the maximizer of

$$\max \; c^T z \quad \text{subject to} \quad Mz \leq \gamma N \tag{3.10}$$

is given by γz^*. Furthermore, the optimal values of the objective functions in (3.9) *and* (3.10) *are given by $c^T z^*$ and $c^T \gamma z^*$. That is, the optimal value of* (3.10) *is γ times larger than that of* (3.9).

Proof. Given (3.9), we rewrite the constraint by multiplying both sides by γ: $M(\gamma z) \leq \gamma N$. Furthermore, we can multiply the cost function by γ, which is permitted because a positive scaling on the objective function of a linear programming problem does not

change the optimizer. We thus obtain the equivalent linear program:

$$\max\ c^T(\gamma z) \quad \text{subject to} \quad M(\gamma z) \leq \gamma N$$

which has the same optimizer as (3.9) but a different objective function value. Finally, we can perform a change of variable $\gamma z \to z$ to transform this optimization into (3.10). It can be concluded that, if the optimizer of (3.9) is z^*, the optimizer of (3.10) must be γz^*.

□

Noting that (2.8) and (3.9) are the same linear program, we can now apply the results of Lemma 1 to the LP in (2.8) to show that admissibility index of $O_\infty^-(r^+)$ in (3.6) is unaffected by a positive scaling on r^+ (the same argument holds true for $O_\infty^+(r^+)$ as well). Specifically, suppose the redundancy of a new inequality $c^T z \leq r^+$ is tested against the partially constructed MAS given by $Mz \leq N$, where $M = [H_x,\ H_v]$, $N = r^+h^-$, and $z = (x,\ v)$. From Lemma 1, scaling r^+ by $\gamma \in \mathbb{R}^+$ (i.e., replacing r^+ with γr^+) scales the optimal solution of the LP by γ. However, the constraint being tested for redundancy is also scaled by γ, which implies that the redundancy of $c^T z \leq \gamma r^+$ is unaffected by γ. Therefore, we conclude that the admissibility index of $O_\infty^-(r^+)$ is unaffected by a positive scaling on r^+.

Now assume $r < 0$, denoted r^-. A polyhedral representation of MAS for the case of $S = 1$ is given by

$$O_\infty^-(r^-) = \left\{(x, v) : H_x x + H_v v \leq r^- h^+\right\} \tag{3.11}$$

Note that we choose h^+ in (3.11) to ensure that the steady-state constraint is indeed

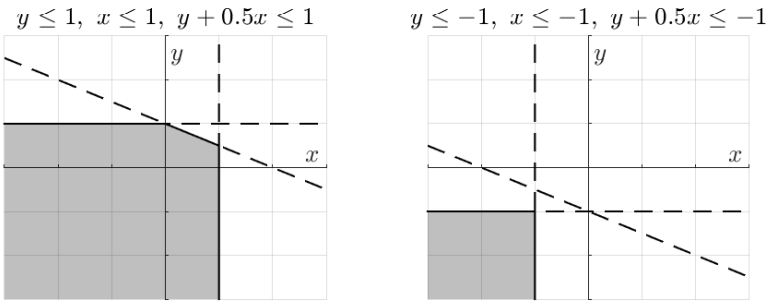

Figure 3.1: Two systems of linear inequalities in $\mathbb{R}^2$: constraints are positive in the left plot and negative in the right plot.

contracted. Similarly, a representation of MAS for the case of $S = -1$ is:

$$O_\infty^+(r^-) = \left\{(x, v) : H_x x + H_v v \geq r^- h^-\right\} \tag{3.12}$$

Using Lemma 1, we can conclude that the admissibility index of $O_\infty^-(r^-)$ in (3.11) is also unaffected by a positive scaling on r^- and that the same argument holds true for $O_\infty^+(r^-)$ in (3.12).

Now suppose that r is allowed to be any non-zero real number. If r changes sign, the geometric properties of the MAS change (graphical argument presented in Fig. 3.1), which in turn changes the linear programs in (2.8) and (3.8). However, with the help of Lemma 2 below, we prove that the admissibility index of (3.6), is equivalent to the admissibility index of (3.12), and similarly that the admissibility index of (3.7), is equivalent to the admissibility index of (3.11).

Lemma 2. *suppose the maximizer of*

$$\max\ c^T z \quad \text{subject to} \quad Mz \leq N \tag{3.13}$$

is given by z^*. Then, the minimizer of

$$\min c^T z \quad \text{subject to} \quad Mz \geq -N \tag{3.14}$$

is given by $-z^*$.

The proof for Lemma 2 is simple and follows similarly to Lemma 1, we thus omit the proof. From Lemma 2, we can conclude that if the maximum value of the objective function in (3.13) is $c^T z^*$, then the minimum of (3.14) is $c^T(-z^*)$. Applying this result to (3.6), we see that we will obtain the same admissibility index as for (3.12). The same can be said between (3.7) and (3.11).

If we combine the information presented in Lemmas 1 and 2, it follows that, for a given r, O_∞^+ and O_∞^- can both be uniquely defined by only two representations of H_x, H_v, and h, which is the novel result of this subsection. This is summarized in Table 3.1, where superscripts $^+$ and $^-$ are used to denote the two possible representations. This table also highlights the relationship between the dynamic MAS, $O_{\infty,tr}\left(r(t),\ y_{tr}(t)\right)$, and the sets $O_\infty^-(r^+)$, $O_\infty^+(r^+)$, $O_\infty^-(r^-)$, and $O_\infty^+(r^-)$ in (3.6), (3.7), (3.11) , and (3.12) respectively.

Note that the MASs in Cases 1 and 4 share the same matrices H_x^-, H_v^-, h^-. Similarly, the MASs in Cases 2 and 3 share the same matrices H_x^+, H_v^+, h^+. Furthermore, these matrices are constant and do not depend on the magnitude of r. Therefore, to construct these matrices, we can make simplifying assumptions on r. Specifically, to compute H_x^-, H_v^-, h^-, we can assume that $r = 1$ and leverage the standard methods presented in Section 2.1.1. Similarly, to compute H_x^+, H_v^+, h^+, we can assume that $r = -1$ and use the standard methods.

	$\{y_{tr} \leq r\}$	$\{y_{tr} > r\}$
$r = r^+ > 0$	Case 1: $O_{\infty,tr} = O_\infty^-(r^+)$ Minimal representation: $\left\{H_x^- x + H_v^- v \leq r h^-\right\}$	Case 2: $O_{\infty,tr} = O_\infty^+(r^+)$ Minimal representation: $\left\{H_x^+ x + H_v^+ v \geq r h^+\right\}$
$r = r^- < 0$	Case 3: $O_{\infty,tr} = O_\infty^-(r^-)$ Minimal representation: $\left\{H_x^+ x + H_v^+ v \leq r h^+\right\}$	Case 4: $O_{\infty,tr} = O_\infty^+(r^-)$ Minimal representation: $\left\{H_x^- x + H_v^- v \geq r h^-\right\}$

Table 3.1: The representation of the dynamic maximal admissible set.

Note that the only remaining case to consider is $r = 0$. In this case, the constraint on the steady state cannot be shrunk (because $(1-\epsilon)r = (1+\epsilon)r = r$ when $r = 0$), resulting in a MAS that may not necessarily be finitely determined. We resolve this by approximating it by the representation with the higher number of rows. This completes the answer to the first question raised in Section 1.2.1 regarding the admissibility index of the dynamic MAS.

We next study the geometric properties of the dynamic MAS as a function of r using Propositions 1 and 2 below.

Proposition 1. *Let $O_\infty^-(r^+)$, $O_\infty^+(r^+)$, $O_\infty^-(r^-)$ and $O_\infty^+(r^-)$ be defined by* (3.6), (3.7), (3.11), *and* (3.12) *respectively; and let $r_1,\ r_2 \in \mathbb{R} \setminus \{0\}$. Then, the following holds.*

i) If $\frac{r_1}{r_2} > 0$, then $O_\infty^-(r_1) = \frac{r_1}{r_2} O_\infty^-(r_2)$ and $O_\infty^+(r_1) = \frac{r_1}{r_2} O_\infty^+(r_2)$

ii) If $\frac{r_1}{r_2} < 0$, then $O_\infty^-(r_1) = \frac{r_1}{r_2} O_\infty^+(r_2)$ and $O_\infty^+(r_1) = \frac{r_1}{r_2} O_\infty^-(r_2)$

Proof. For clarity throughout the proof, let superscripts $^+$ (positive) and $^-$ (negative)

denote the signs of r_1 and r_2. We prove case *i*) and *ii*) for $O_\infty^-(r_1^+)$; the rest of the cases can be proven similarly.

i) Let $(x, v) \in O_\infty^-(r_1^+)$. Then, it follows from (3.6) that

$$\begin{aligned} & H_x x + H_v v \leq r_1^+ h^- \iff \frac{r_2^+}{r_1^+}\left(H_x x + H_v v\right) \leq \frac{r_2^+}{r_1^+}\left(r_1^+ h^-\right) \\ \iff & H_x\left(\frac{r_2^+}{r_1^+}x\right) + H_v\left(\frac{r_2^+}{r_1^+}v\right) \leq r_2^+ h^- \iff \frac{r_2^+}{r_1^+}(x, v) \in O_\infty^-(r_2^+) \\ \iff & (x, v) \in \frac{r_1^+}{r_2^+} O_\infty^-(r_2^+) \end{aligned}$$

ii) Let $(x, v) \in O_\infty^-(r_1^+)$. Then, it follows from (3.6) that

$$\begin{aligned} & H_x x + H_v v \leq r_1^+ h^- \iff \frac{r_2^-}{r_1^+}\left(H_x x + H_v v\right) \geq \frac{r_2^-}{r_1^+}\left(r_1^+ h^-\right) \\ \iff & H_x\left(\frac{r_2^-}{r_1^+}x\right) + H_v\left(\frac{r_2^-}{r_1^+}v\right) \geq r_2^- h^- \iff \frac{r_2^-}{r_1^+}(x, v) \in O_\infty^+(r_2^-) \\ \iff & (x, v) \in \frac{r_1^+}{r_2^-} O_\infty^+(r_2^-) \end{aligned}$$

The remaining 6 cases

$$O_\infty^-(r_1^-),\ O_\infty^+(r_1^+),\ O_\infty^+(r_1^-) \text{ for both } i) \text{ and } ii)$$

can be proven similarly. Furthermore, the reverse direction of each case can be proven.

□

Proposition 1 sheds light on the geometric relationship between O_∞^+ and O_∞^-. For example, for positive values of r, $O_\infty^-(r)$ is scaled radially from the origin as r varies.

Another important result, which ties into recursive feasibility of the RG-DC as addressed in Subsection 3.1.2, is as follows.

Proposition 2. *Suppose $r_2 \geq r_1$, then $O_\infty^-(r_1) \subseteq O_\infty^-(r_2)$. Similarly, if $r_2 \leq r_1$, then $O_\infty^+(r_1) \subseteq O_\infty^+(r_2)$.*

Proof. We prove the first statement of Proposition 2. The second statement follows similarly.

Let $r_2 \geq r_1$, and $(x, v) \in O_\infty^-(r_1)$. We first consider the case where $r_1 \geq 0$, denoted r_1^+, and where $r_2 \geq 0$, denoted r_2^+. From (3.6), it is true that $H_x x + H_v v \leq r_1^+ h^-$. Therefore, because $r_2^+ h^- \geq r_1^+ h^-$, $H_x x + H_v v \leq r_2^+ h^-$. From here it can be concluded that any $(x, v) \in O_\infty^-(r_1^+)$ also belongs to $O_\infty^-(r_2^+)$. Therefore, $O_\infty^-(r_1^+) \subseteq O_\infty^-(r_2^+)$. Now let us consider the case where $r_1 \leq 0$, denoted r_1^-, and where $r_2 \geq 0$, denoted r_2^+. Clearly, $r_2^+ \geq r_1^-$. Let $(x, v) \in O_\infty^-(r_1^-)$. From (3.11), it is true that $H_x x + H_v v \leq r_1^- h^+$. Therefore, because $r_2^+ h^- \geq r_1^- h^+$, $H_x x + H_v v \leq r_2^+ h^-$. From here it can be concluded that any $(x, v) \in O_\infty^-(r_1^-)$ also belongs to $O_\infty^-(r_2^+)$. Therefore, $O_\infty^-(r_1^-) \subseteq O_\infty^-(r_2^+)$. Finally, we consider the case where $r_1 \leq 0$, denoted r_1^-, and where $r_2 \leq 0$, denoted r_2^-. From (3.11), it is true that $H_x x + H_v v \leq r_1^- h^+$. Therefore, because $r_2^- h^+ \geq r_1^- h^+$, $H_x x + H_v v \leq r_2^- h^+$. From here it can be concluded that any $(x, v) \in O_\infty^-(r_1^-)$ also belongs to $O_\infty^-(r_2^-)$. Therefore, $O_\infty^-(r_1^-) \subseteq O_\infty^-(r_2^-)$.

The three cases for the second statement of Proposition 2, regarding $r_2 \leq r_1$, can be proven similarly.

□

Note that $O_\infty^-(r_1) \not\subseteq O_\infty^-(r_2)$ if $r_2 < r_1$, and $O_\infty^+(r_1) \not\subseteq O_\infty^+(r_2)$ if $r_2 > r_1$. This result implies that, while the dynamic MAS is positively invariant for a fixed r, it may not be positively invariant if r varies over time (conditions for positive invariance under time-varying r are given in Proposition 2). The implication of this in terms of constraint management will be discussed in the next subsection. The above

two propositions provide the answer to the second question raised in Section 1.2.1 regarding the geometric properties of the dynamic MAS.

3.1.2 Computational aspects and properties of RG-DC

To implement the RG-DC, the values of $y_{tr}(t)$ and $r(t)$ are used at every timestep to determine the appropriate MAS from Table 3.1. This MAS is then employed in (2.11) to calculate κ. We denote the resulting solution by κ_{tr}. If static constraints are also imposed on the system, we compute (2.11) separately with $O_{\infty,st}$, yielding κ_{st}. The RG-DC then chooses the minimum of $\{\kappa_{tr}, \kappa_{st}\}$ and applies the solution to (2.10) to compute $v(t)$.

As discussed in the previous subsection, the dynamic MAS may or may not be positively invariant if r changes in real-time (conditions for positive invariance were provided in Proposition 2). This implies that the LP in (2.11) may be infeasible, which means that κ_{tr} may not exist. The traditional reference governor handles this situation by forcing κ to be 0 (i.e., $v(t) = v(t-1)$). RG-DC handles infeasibilities in the same manner. Specifically, if at the current timestep the LP in (2.11) is infeasible, we set $\kappa = 0$. In such cases, overshoot is not preventable. However, we maintain $\kappa_{tr} \in [0, 1]$ in our RG-DC formulation to assure stability at the expense of overshoot mitigation performance. We demonstrate a scenario where the RG-DG forces $\kappa = 0$ in Section 3.2.

We discuss the stability of the RG-DC loop in Theorem 1 below.

Theorem 1. *The RG-DC loop is BIBO stable, and for a constant r, v converges to a constant.*

Proof. From (2.10), and with $\kappa \in [0, 1]$, it follows that $v(t)$ is a convex combination of $r(t)$ and $v(t-1)$, both of which are bounded. Therefore, $v(t)$ is bounded. Boundedness of $v(t)$ and asymptotic stability of (3.1) imply BIBO stability of the system. Furthermore, $v(t)$ forms a monotonic sequence bounded by r, which implies convergence. □

Note that this result is similar to the stability result of the standard RG. However, we present it formally to reinforce the claim that, like the RG, the RG-DC is BIBO stable.

The computational complexity of the RG-DC is similar to that of the standard RG with an additional constraint on the tracking output. Note that the additional logic introduced to determine the MAS characterization from Table 3.1 is negligible when compared to the processing times associated with the calculation of κ in (2.11). The RG-DC is also comparable to the RG in terms of memory requirements.

Finally, note that external disturbances, model uncertainty, and plant variability can be naturally incorporated in the RG-DC framework. This is done, similar to standard RG, by "robustifying" (i.e., shrinking) the MAS using the ideas from Pontryagin subtraction (P-subtraction) [12] and polytopic uncertainties [29]. We will show an example of this in the next section.

The above analyses provide complete answers to questions 3 – 6 raised in Section 1.2.1 regarding the properties of RG-DC.

We now present the RG-DC algorithm (see Algorithm 3), which can be used to enforce overshoot mitigation constraints using $O_{\infty,tr}$ and static constraints using $O_{\infty,st}$. In preparation for Algorithm 3, assume that the two representations of the dynamic MAS, namely H_x^-, H_v^-, h^- and H_x^+, H_v^+, h^+, have been calculated. Let $H_{x,tr}, H_{v,tr}, h_{tr}$ be the representation with the larger number of rows, where h_{tr} is a vector of all 1s. Additionally, let $H_{x,st}, H_{v,st}, h_{st}$ denote the matrices that define $O_{\infty,st}$. The RG-DC algorithm is presented in Algorithm 3.

Algorithm 3 RG-DC

Inputs:
$y(t), r(t), x(t), v(t-1), H_{x,tr}, H_{v,tr}, h_{tr}, H_{x,st}, H_{v,st}, h_{st}, \epsilon$
Output:
$v(t)$

initialize j as 1
initialize κ as a vector with the same number of rows as h_{tr}
if $y_{tr}(t) \leq r(t)$ **then**
    if $r(t) > 0$ **then**
        assign the first row of h_{tr} as $(1 - \epsilon)$
    else
        assign the first row of h_{tr} as $(1 + \epsilon)$
    end if
    for each row in $0_{\infty,tr}$ **do**
        $n := h_{tr}(\text{row } j)r(t) - H_{x,tr}(\text{row } j)x(t) - H_{v,tr}(\text{row } j)v(t-1)$
        $d := H_{v,tr}(\text{row } j)(r(t) - v(t-1))$
        $\kappa(\text{row } j) := \text{kappa}(n, d)$
        $j := j + 1$
    end for
else
    if $r(t) > 0$ **then**
        assign the first row of h_{tr} as $(1 + \epsilon)$
    else
        assign the first row of h_{tr} as $(1 - \epsilon)$
    end if
    for each row in $0_{\infty,tr}$ **do**
        $n := -h_{tr}(\text{row } j)r(t) + H_{x,tr}(\text{row } j)x(t) + H_{v,tr}(\text{row } j)v(t-1)$
        $d := -H_{v,tr}(\text{row } j)(r(t) - v(t-1))$
        $\kappa(\text{row } j) := \text{kappa}(n, d)$
        $j := j + 1$
    end for
end if
$\kappa_{tr} := \min(\kappa)$
if there are any static constraints **then**
    use standard RG algorithm with $O_{\infty,st}$ to obtain κ_{st}
else
    $\kappa_{st} := 1$
end if
$\kappa^* := \min(\kappa_{tr}, \kappa_{st})$
$v(t) := v(t-1) + \kappa^*(r(t) - v(t-1))$

function kappa(n, d)

if $n > 0$ **then**
    if $d > 0$ **then**
        $\kappa := \min(n/d, 1)$
    else
        $\kappa := 1$
    end if
else
    $\kappa := 0$
end if
return κ

end function

3.2 Illustrative Examples

3.2.1 System model

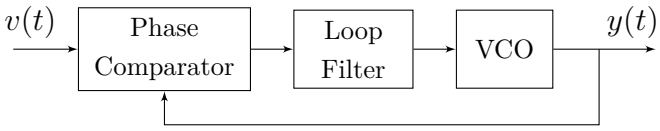

Figure 3.2: Simple analog PLL system.

Consider the analog phase locked loop (PLL) system shown in Fig. 3.2, which is comprised of a phase comparator, a loop filter, and a voltage controlled oscillator (VCO). The transfer function of the closed-loop PLL system around a nominal operating point is as follows [65]:

$$H_{PLL} = \frac{G_{lp}G_{VCO}}{s^2 + G_{lp}s + G_{lp}G_{VCO}} \tag{3.15}$$

where G_{lp} is the loop filter parameter and G_{VCO} is the VCO gain. Note that the closed-loop system has a DC gain of 1 and perfect steady-state tracking of step commands. For the simulation, G_{lp} was chosen to be 100, and G_{VCO} was chosen to be $2G_{lp}$ to yield an underdamped system with damping ratio $\zeta = 0.35$. By selecting states as $x_1 = y$ and $x_2 = \dot{y}$, a zero order hold discretization of the system with a sample time of $T_s = 1 \times 10^{-4}$ seconds is used to obtain the discrete state-space model of the closed-loop system.

Constraints are imposed on both outputs of the system. The dynamic constraint is applied to the tracking output $y_{tr} \triangleq y_1$ and a slew-rate limiting constraint ($-100 \leq$

$y_2 \leq 100$) is applied to the constrained output $y_{st} \triangleq y_2$. The static and dynamic maximal admissible sets are constructed as discussed in Sections 2.1.1 and 3.1. The resulting polyhedra, $O_\infty^-(r^+)$ and $O_\infty^-(r^-)$, both have admissibility indices of 342 (the representations happen to be the same for this example). Additionally, the admissibility index of $O_{\infty,st}$ is 130.

3.2.2 Response Evaluation

Fig. 3.3 shows the improved response characteristics of the governed system compared to the ungoverned system. Note that overshoot was completely eliminated without making any modifications to the PLL. Hence, the RG-DC is especially effective in overshoot mitigation of systems with inner loop controllers that cannot be tuned or adjusted (i.e 'black box' systems), which is true for many off-the-shelf PLLs.

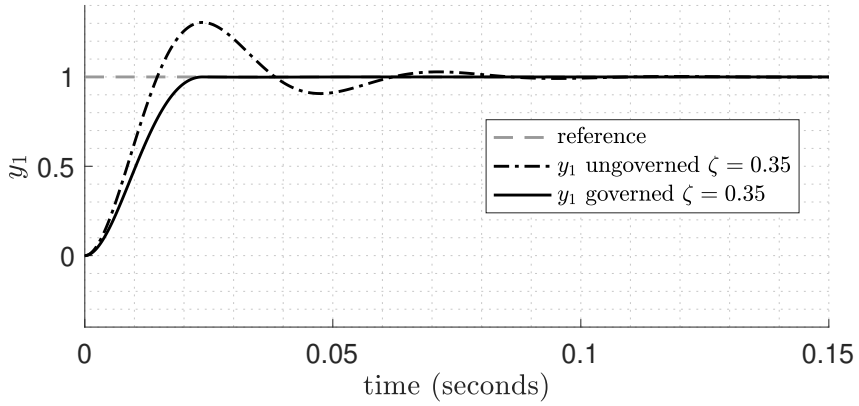

Figure 3.3: Comparison between step responses of the governed and ungoverned systems. No slew-rate limit is applied to the governed system in this simulation.

A simulation of the RG-DC operating on step signals is presented in Fig. 3.4. Notice that overshoot is mitigated for all but the last step at $t = 0.208$ seconds. In this case, the reference changes quickly so that $(x(t), v(t-1))$ does not belong to the

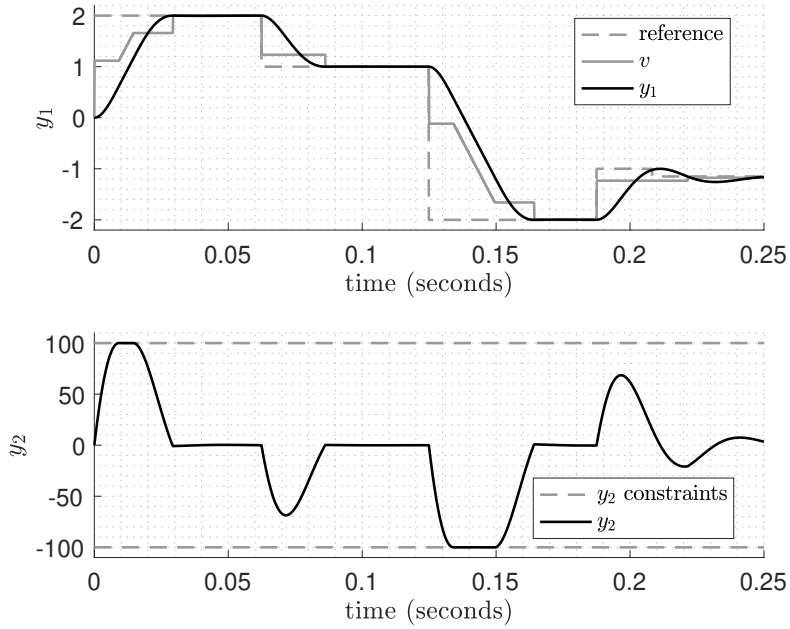

Figure 3.4: Governed system response to multiple step inputs (slew-rate limit = 100)

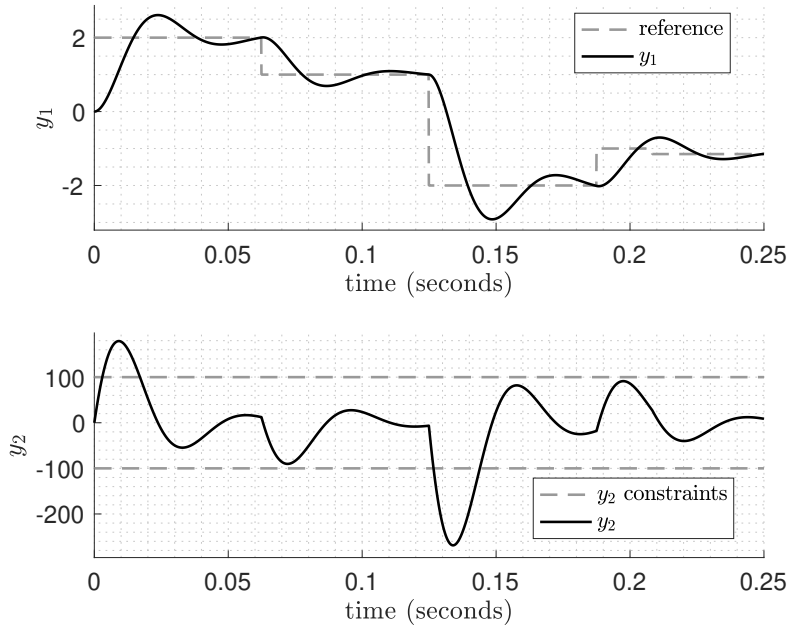

Figure 3.5: Ungoverned system response to multiple step inputs

new MAS, which means constraint violation is not preventable. Hence, κ has been set to 0. Note that we maintain convergence to the reference at the sacrifice of reduced overshoot mitigation performance. For comparison, the ungoverned system response to the same reference signal is shown in Figure 3.5.

3.2.3 Robustness

To test robustness under model uncertainty, we treat the VCO gain, G_{VCO}, as an unknown. We suppose, however, that G_{VCO} is bounded as follows: $160 \leq G_{VCO} \leq 240$. We compute a robust MAS for this system using Algorithm 1 from [29]. Fig. 3.6 compares the robust MAS with a standard MAS generated with the nominal model parameter $G_{VCO} = 200$. From the figure, it is evident that the introduction of model uncertainty results in a more conservative MAS. Upon simulation of the governed system with the robust MAS, we see in Fig. 3.7 that the constraints are not violated for systems at the vertices of system uncertainty (the values of G_{VCO} used for the simulations are shown in the figure titles).

3.2.4 RG-DC as a nonlinear filter

Finally, we present an interesting experiment, which led to a thought-provoking observation regarding the frequency response of the governed PLL system, which we discuss next.

In [66], it is shown that nonlinear systems can be analyzed using frequency domain techniques if they satisfy the “convergence" property. Essentially, a system is defined to be convergent if, akin to a linear system, its response converges to the forced

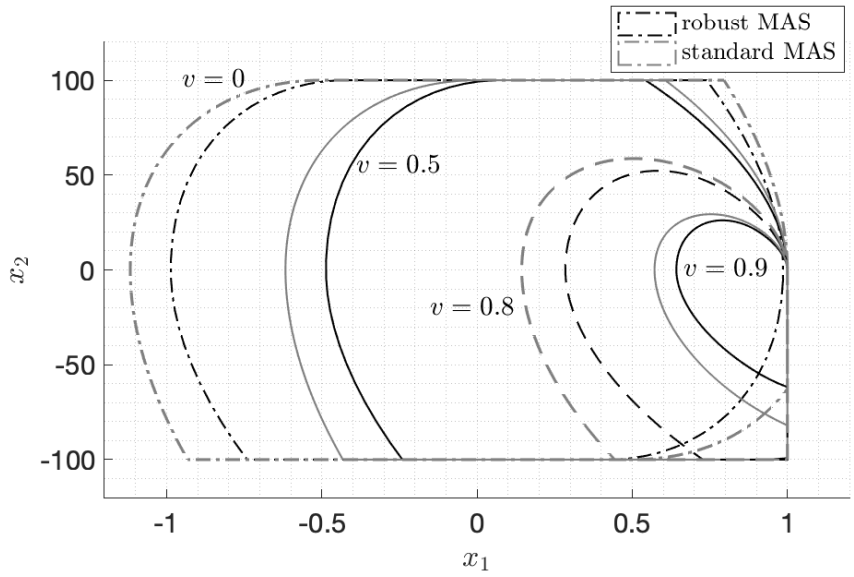

Figure 3.6: Slices from the robust and standard MASs at various values of v. The dynamic constraint for this plot is $y_1 \leq 1$ and the slew-rate limit is 100.

response, regardless of the initial conditions. As argued in [66], nonlinear convergent systems can be analyzed using the *nonlinear Bode magnitude plot*, which is a proper extension of the traditional Bode magnitude plot for linear systems. However, unlike the linear Bode plot, which is only a function of the frequency of the input sinusoid, the nonlinear Bode plot is generally a function of both the frequency and amplitude of the input.

In our case, it can be shown that the overall system with RG-DC governing the input is indeed a convergent system (see Fig. 3.8 for graphical argument). Furthermore, as we prove in Theorem 2, the RG-DC is such that the nonlinear Bode plot has no dependence on the amplitude of the input because the system satisfies the homogeneity condition, similar to a linear system. Thus, we adopt the methods in [66] to generate a nonlinear Bode magnitude plot of the governed system as a function of the input frequency only. This plot is presented in Fig. 3.9, which also shows the Bode

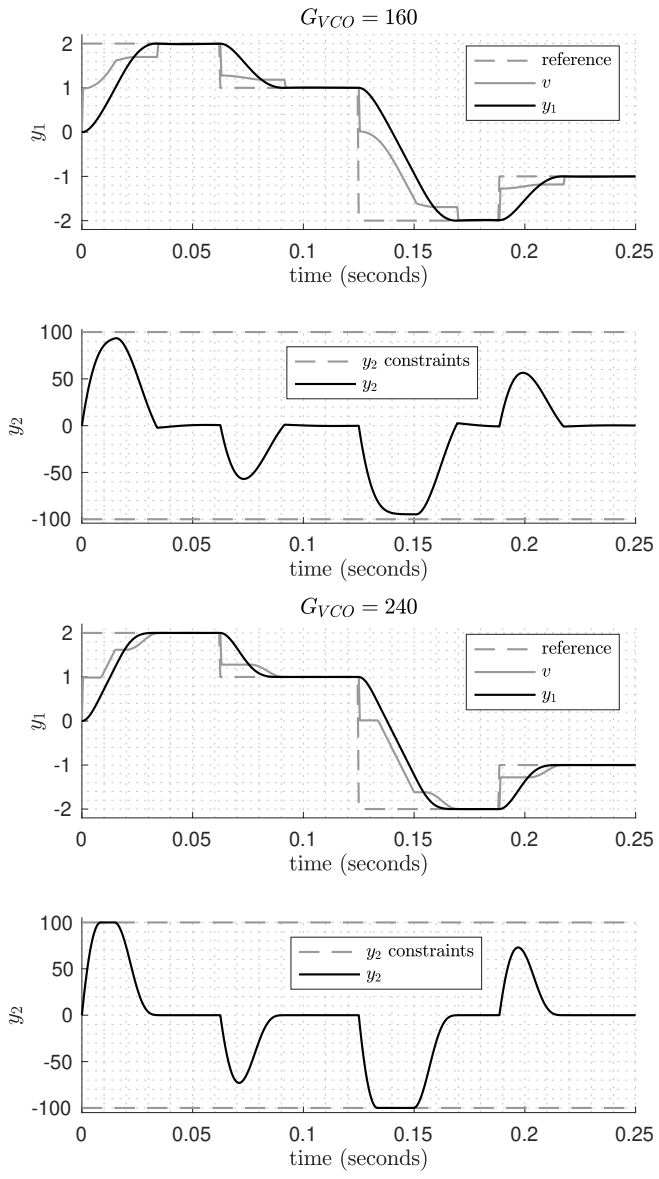

Figure 3.7: Governed uncertain system responses to multiple step inputs (slew-rate limit = 100*). Note that the top two sub-figures and the bottom two sub-figures correspond to two different realizations of the system uncertainty shown in the plot titles.*

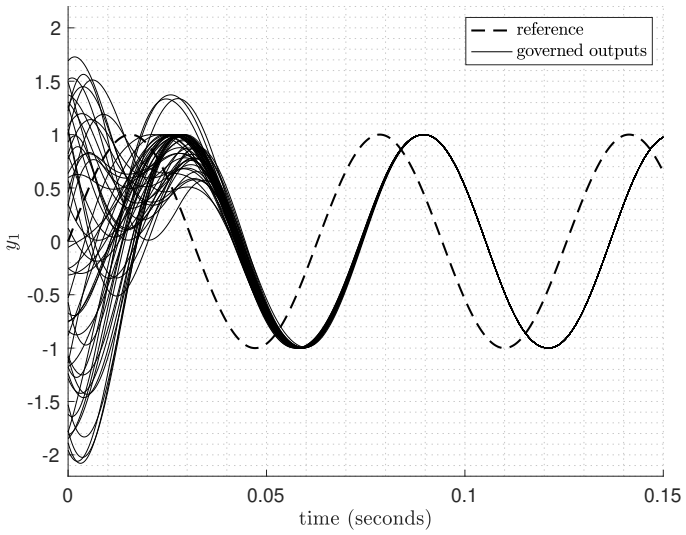

Figure 3.8: Demonstration of convergence via simulation of the governed PLL system (no slew-rate limit) at 50 *jointly uniformly distributed random initial conditions* (x_0, v_0)*. Initial condition ranges:* $x_{0_1} \in [-2, 2]$, $x_{0_2} \in [-200, 200]$, $v_0 \in [-1, 1]$*. The reference* $r(t)$ *is a sinusoid with frequency* 100 *rad/s. Note that overshoot mitigation constraints for some initial conditions were not satisfied because the initial conditions did not belonged to MAS.*

magnitude plot of the ungoverned PLL system (3.15) as comparison. The other plots labeled "2^{nd} order system" and "12^{th} order system" will be explained later. Details on how each plot was generated can be found in the caption of Fig. 3.9. Upon inspection of Fig. 3.9, it appears that the resonant peak inherent in the Bode magnitude plot of the underdamped closed loop PLL system is completely eliminated with the implementation of the RG-DC. These results suggest that RG-DC could potentially be used in conjunction with a resonant low-pass filter, giving the ability to eliminate the resonant behavior without greatly affecting the cutoff frequency or the attenuation properties beyond the cutoff frequency. In other words, the RG-DC may be thought of as a "nonlinear" filter with anti-resonance properties.

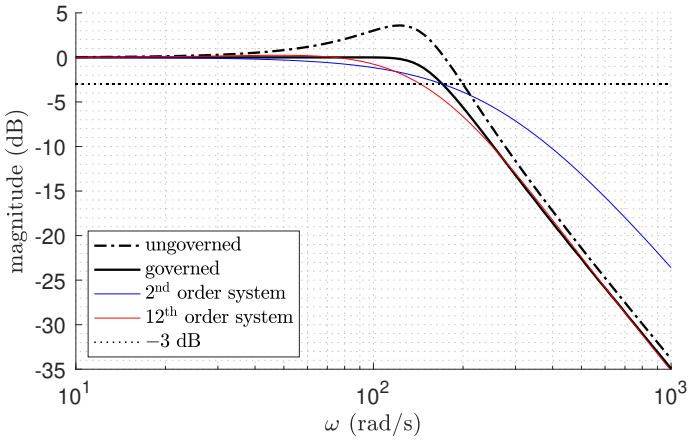

Figure 3.9: Bode magnitude plot (ungoverned PLL system) and nonlinear Bode magnitude plot (governed PLL system, no slew-rate limit). In addition, the Bode magnitude plots of the 2^{nd} *order and* 12^{th} *order systems are shown. The nonlinear Bode magnitude plot was generated by simulating governed system responses with sinusoidal references of amplitude* 1 *at* 100 *different frequencies that were logarithmically equally spaced ranging from* 10 *rad/s to* $1,000$ *rad/s. The supremum norm of the outputs were measured at steady-state and converted to dB. The ungoverned PLL system,* 2^{nd} *order system, and* 12^{th} *order system Bode magnitude plots were generated using the standard linear systems approach applied to the respective linear system models.*

We highlight the fact that the resonant peak in the Bode magnitude plot of the ungoverned system shown in Fig. 3.9 is caused by underdamped poles in (3.15). Therefore, a natural solution to remove the resonant peak is by using plant inversion, i.e., replacing the RG-DC in Fig. 1.1 with an inverse model that cancels the underdamped poles of the closed-loop system with a pair of complex conjugate zeros. Since the resulting inverse model would be improper, additional (overdamped) poles must be added to obtain a proper transfer function. The series connection of the inverse model and the closed-loop system yields an equivalent transfer function with no resonance peaks. This raises the following question: how does the nonlinear Bode magnitude plot from the governed PLL system compare with the linear Bode magnitude plot of this equivalent system? To have a fair comparison, we introduce

two choices for the equivalent systems, both with DC gain equal to 1 and relative degree equal to 2 to maintain the -40 dB/decade roll-off of the governed PLL system. The first system, of order 2, was designed such that the -3 dB bandwidth was equal to that of the governed PLL system (based off the nonlinear Bode magnitude plot), whereas the second system, of order 12, was designed to minimize the root-mean-square error relative to the nonlinear Bode magnitude plot. The results are shown in Fig. 3.9. Note that the frequency response of the governed PLL system has a faster transition from 0 to -40 dB/decade compared to the frequency response of the 2^{nd} order system. Furthermore, the 12^{th} order Bode magnitude plot matches the nonlinear Bode magnitude plot except for the frequency range from 100 rad/s to 200 rad/s where the magnitude of the 12^{th} order frequency response is less than the magnitude of the governed frequency response. We thus make the mild conclusion that the frequency response provided by the governed PLL system is not attainable by a low-order linear system (of degree less than 12), which shows that an RG-DC governed resonant low-pass filter does indeed produce a novel frequency response.

We now prove the homogeneity property of the overall system with the RG-DC governing the reference, as alluded to above. In preparation for Theorem 2, we introduce the following notation. Let the governed output of system (3.1) be $y_{tr}\left(t, r(t), (x_0, v_0)\right)$, where $r(t)$ is the reference signal that is applied to the system depicted in Fig. 1.1 with initial conditions $(x_0, v_0) := (x(0), v(-1))$ belonging to MAS. The following theorem holds.

Theorem 2. *Suppose $p = 0$ in system (3.1), so that $y_{tr}(t)$ is the only output governed by RG-DC. Then $y_{tr}\left(t, \alpha r(t), \alpha(x_0, v_0)\right) = \alpha y_{tr}\left(t, r(t), (x_0, v_0)\right)$, $\forall \alpha \in \mathbb{R}^+$.*

Proof. We prove the homogeneity condition of the RG-DC from r to v by principal of

induction. The homogeneity condition from r to y_{tr} then follows from the fact that the closed-loop system (3.1) from v to y_{tr} is linear.

We first establish the base case of the inductive argument, where we prove that scaling the initial conditions, $(x_0, v_0) = (x(0), v(-1))$, and the reference, $r(0)$, by α (written in short by $(x_0, v_0) \to \alpha(x_0, v_0)$, $r(0) \to \alpha r(0)$), scales the next iterate by α: $(x(1), v(0)) \to \alpha(x(1), v(0))$. To show this, consider the RG algorithm from (2.11) at time $t = 0$ with the change of variables: $x(0) \to \alpha x(0)$, $v(-1) \to \alpha v(-1)$, and $r(0) \to \alpha r(0)$. Finally, let $\widetilde{O}_\infty$ from (2.11) be $O_{\infty,tr}\left(\alpha r(0), \alpha y_{tr}(0)\right) = O_{\infty,tr}\left(\alpha r(0), \alpha C_{tr} x_0\right)$. The optimization problem becomes:

$$\begin{aligned} \underset{\kappa \in [0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & v(0) = \alpha v_0 + \kappa\left(\alpha r(0) - \alpha v_0\right) \\ & \left(\alpha x_0,\ v(0)\right) \in O_{\infty,tr}\left(\alpha r(0), \alpha C_{tr} x_0\right) \end{aligned} \tag{3.16}$$

Recall from Table 3.1, that the relationship between $y_{tr}(t)$ and $r(t)$ determines which of the four cases of dynamic MAS is used in the RG algorithm at timestep t. Furthermore, note that scaling $r(t)$ and $y_{tr}(t)$ by α does not change which case is appropriate. This implies that $O_{\infty,tr}\left(r(t), C_{tr} x(t)\right)$ and $O_{\infty,tr}\left(\alpha r(t), \alpha C_{tr} x(t)\right)$, at any instance in time, both belong to the same case from Table 3.1. By Proposition 1 i), it then follows that $O_{\infty,tr}\left(\alpha r(0), \alpha C_{tr} x_0\right) = \alpha O_{\infty,tr}\left(r(0), C_{tr} x_0\right)$, and we can conclude that the constraints of optimization problem (3.16) are unaffected by α. Furthermore, noting that the cost function of (3.16) also does not depend on α, it follows that optimization problem (3.16) results in the same optimizer κ^* at time $t = 0$ regardless of α. From here, it follows that the modified reference from (3.16) is $\alpha v(0)$. Furthermore, at time $t = 1$ we can conclude that $\alpha x(1) = A\alpha x_0 + B\alpha v(0)$. Note that this base case also

holds when there is no solution to the optimization problem because $\kappa^* = 0$ at time $t = 0$ means that $\alpha v(0) = \alpha v_0$.

We now present the induction step, where we prove that scaling the parameters, $x(t)$, $v(t-1)$, and $r(t)$, by α, gives the following result: $v(t) \to \alpha v(t)$ and $x(t+1) \to \alpha x(t+1)$. Consider the RG algorithm from (2.11) with the change of variables: $x(t) \to \alpha x(t)$, $v(t-1) \to \alpha v(t-1)$, and $r(t) \to \alpha r(t)$. Finally, let $\widetilde{O}_\infty$ from (2.11) be $O_{\infty,tr}\left(\alpha r(t), \alpha y_{tr}(t)\right) = O_{\infty,tr}\left(\alpha r(t), \alpha C_{tr} x(t)\right)$. The optimization problem becomes:

$$\begin{aligned} \underset{\kappa \in [0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & v(t) = \alpha v(t-1) + \kappa\left(\alpha r(t) - \alpha v(t-1)\right) \\ & \left(\alpha x(t),\ v(t)\right) \in O_{\infty,tr}\left(\alpha r(t), \alpha C_{tr} x(t)\right) \end{aligned} \tag{3.17}$$

Again, note that because $O_{\infty,tr}\left(\alpha r(t), \alpha C_{tr} x(t)\right) = \alpha O_{\infty,tr}\left(r(t), C_{tr} x(t)\right)$, the constraints of optimization problem (3.17) are independent of α. Furthermore, noting that the cost function of (3.17) also does not depend on α, it follows that (3.17) results in the same optimizer κ^* at time t regardless of α. From here, it follows that the modified reference from (3.17) is $\alpha v(t)$. Furthermore, at time $t+1$ we can conclude that $\alpha x(t+1) = A\alpha x(t) + B\alpha v(t-1)$. Note that this induction step also holds when there is no solution to the optimization problem because $\kappa^* = 0$ at time t means that $\alpha v(t) = \alpha v(t-1)$. Considering the above logic, by the principal of induction, the RG-DC satisfies the homogeneity condition (from r to v) and we can conclude that, because the closed-loop system is linear (from v to y_{tr}), the entire governed system (from r to y_{tr}) satisfies the homogeneity condition. This completes the proof. □

An additional argument can be made that further strengthens the validity of the nonlinear Bode magnitude plot. We argue that although the input to the linear closed-loop system ($v(t)$) is not perfectly sinusoidal (due to the governing action of the RG-DC), the output of the system (y_{tr}) is "approximately" sinusoidal (which we have observed in our simulations, see for example Fig. 3.8). The reason for this phenomenon can be attributed to the fact that the closed-loop system is of low-pass-filtering nature, which implies that it filters out higher order harmonics of $v(t)$. This argument is similar to the methods used for Describing Functions [67].

Chapter 4

RG-AC: Reference Governor for Slowly Time-Varying Systems and Constraints

As mentioned in the Introduction, the Adaptive-Contractive Reference Governor (RG-AC) is derived from the Reference Governor (RG) framework and utilizes a contractive constraint set, along with a dynamic approximation of the Maximal Admissible Set (MAS) to handle constraint management under time-varying plant dynamics and time-varying constraints. In this chapter we develop RG-AC by first introducing two unique methods of MAS characterization. The first method, called Parameter-Dependent MAS, assumes a linear parameter varying (LPV) system model, where the elements of the A, B, C, and D matrices may be functions of one or more real valued parameters. In this formulation, we linearize the parameter-dependent matrices characterizing MAS around some nominal parameter values. This allows us to approximate the change in the edges of Parameter-Dependent MAS as the parameters vary around the nominal values. The second method, called λ-contractive MAS, as proposed in [23], develops MAS that, in addition to being invariant, is also contractive in cross-sections on $v(t)$. This formulation allows for constraint management of slowly time-varying constraints by guaranteeing membership of the constrained outputs to a contractive constraint set. To complete the chapter, we introduce the RG-AC algorithm which utilizes a λ-contractive, Parameter-Dependent MAS to compute $v(t)$ and safely and effectively govern systems with slowly time-varying dynamics and slowly time-varying constraints.

4.1 Parameter-Dependent MAS

In this section we introduce the novel Parameter-Dependent MAS, which is one of the two unique methods used in the formulation of MAS for RG-AC. To elaborate, consider the system:

$$\begin{aligned} x(t+1) &= A(\Theta)x(t) + B(\Theta)v(t) \\ y(t) &= C(\Theta)x(t) + D(\Theta)v(t) \end{aligned} \tag{4.1}$$

where $v(t+1) = v(t)$ and $y(t)$ is subject to the convex polyhedral constraints:

$$y(t) \in \mathbb{Y} := \{y : Sy(t) \leq s\} \tag{4.2}$$

where $\Theta \in \mathbb{R}^q$ is a parameter on which the elements of A, B, C, D depend on, and for now is not assumed to vary with respect to time. Additionally, $s \in \mathbb{R}^b,\ s \geq 0$; and $\mathbb{Y}$ is nonempty, compact, and contains zero. We denote the nominal system parameters as Θ_n, and assume there exists a closed neighborhood of Θ_n denoted by the set $\mathbb{T} = \{\Theta : \|\Theta - \Theta_n\| \leq R\}$ where $R \in \mathbb{R}^+$, such that the elements of A, B, C, D are uniformly continuous in Θ, the pair $(A(\Theta), C(\Theta))$ is observable, and (4.1) is asymptotically stable. For the remainder of this chapter, we assume that $\Theta \in \mathbb{T}$.

Recall when forming MAS, that $v(t)$ is held constant for all time ($v(t+1) = v(t)$). Thus, (4.1) may be written in the form:

$$\begin{aligned} \bar{x}(t+1) &= \bar{A}(\Theta)\bar{x}(t) \\ y(t) &= \bar{C}(\Theta)\bar{x}(t) \end{aligned} \tag{4.3}$$

where

$$\bar{x}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}, \quad \bar{A}(\Theta) = \begin{bmatrix} A(\Theta) & B(\Theta) \\ 0 & I \end{bmatrix}, \quad \bar{C}(\Theta) = \begin{bmatrix} C(\Theta) & D(\Theta) \end{bmatrix}$$

With all eigenvalues associated with I being simple, (4.3) is clearly Lyapunov stable for all $\Theta \in \mathbb{T}$.

Let $\widetilde{O}_\infty(\Theta) = \{\bar{x} : H(\Theta)\bar{x} \leq h\}$ denote the MAS associated with system (4.3) (note that, in general, $\widetilde{O}_\infty(\Theta)$ is parameter-dependent) and let $\widetilde{O}_\infty(\Theta_n) = \{\bar{x} : H_n\bar{x} \leq h\}$ denote the MAS associated with system (4.3) for $\Theta = \Theta_n$ (note that $\widetilde{O}_\infty(\Theta_n)$ is a specific instance of $\widetilde{O}_\infty(\Theta)$). Our goal is to characterize MAS for system (4.3) as a function of Θ without total recalculation, which is not computationally efficient and may not be practical if calculation times are limited. We achieve this by using a truncated Taylor series of $H(\Theta)$ expanded around $\Theta = \Theta_n$. Assuming that the change of Θ around Θ_n is small, we only keep the linear term, resulting in a first-order Taylor approximation. Thus, Parameter-Dependent MAS is based on a linear approximation with respect to Θ of $\widetilde{O}_\infty(\Theta)$ at $\Theta = \Theta_n$ (the approximation is hereafter referred to as $\widetilde{O}_{\infty,T}(\Theta) = \{\bar{x} : H_T(\Theta)\bar{x} \leq h\}$). Note that the terms "linear approximation" and "first-order Taylor approximation" are used interchangeably for the remainder of this thesis. To demonstrate the computation of $\widetilde{O}_{\infty,T}(\Theta)$, suppose that $\widetilde{O}_\infty(\Theta_n)$ has been calculated and is represented by the polytope $\{\bar{x} : H_n\bar{x} \leq h\}$. The calculation of the matrix $H_T(\Theta)$, which is a first-order Taylor approximation of $H(\Theta)$ at $\Theta = \Theta_n$, and is the only parameter-dependent portion of $\widetilde{O}_{\infty,T}(\Theta)$, is as follows:

$$H(\Theta) \approx H_T(\Theta) = H_n + H_s(\Theta) \tag{4.4}$$

where

$$H_s(\Theta) = \sum_{i=1}^{q} \frac{\partial H(\Theta)}{\partial \Theta_i}\bigg|_{\Theta_n} (\Theta_i - \Theta_{n,i}). \tag{4.5}$$

Note that Θ_i denotes the i^{th} element of Θ, and $\Theta_{n,i}$ denotes the i^{th} element of Θ_n. For completeness, we cover the analytical computation of $\frac{\partial H(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$. First, suppose that $\widetilde{O}_\infty(\Theta_n)$ has been calculated using the standard methods (see Algorithm 1) and has an admissibility index denoted by $j^*_{\Theta_n}$. We express $H(\Theta)$, built up to $(j^*_{\Theta_n} + 2)b$ rows as follows:

$$H(\Theta) = \begin{bmatrix} \lim\limits_{j\to\infty} S\bar{C}(\Theta)\bar{A}(\Theta)^j \\ S\bar{C}(\Theta) \\ S\bar{C}(\Theta)\bar{A}(\Theta) \\ S\bar{C}(\Theta)\bar{A}(\Theta)^2 \\ \vdots \\ S\bar{C}(\Theta)\bar{A}(\Theta)^{j^*_n} \end{bmatrix} \tag{4.6}$$

where

$$\lim_{j\to\infty} S\bar{C}(\Theta)\bar{A}(\Theta)^j = \begin{bmatrix} 0 & S\Big(C(\Theta)(I - A(\Theta))^{-1}B(\Theta) + D(\Theta)\Big) \end{bmatrix} \tag{4.7}$$

The partial derivatives of (4.6) with respect to Θ_i are

$$\frac{\partial H(\Theta)}{\partial \Theta_i} = \begin{bmatrix} \lim\limits_{j\to\infty} S\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)^j}{\partial \Theta_i} \\ S\frac{\partial \bar{C}(\Theta)}{\partial \Theta_i} \\ S\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)}{\partial \Theta_i} \\ S\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)^2}{\partial \Theta_i} \\ \vdots \\ S\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)^{j^*_n}}{\partial \Theta_i} \end{bmatrix} \tag{4.8}$$

where

$$\lim_{j\to\infty} S\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)^j}{\partial \Theta_i} = \begin{bmatrix} 0 & S\Big(\frac{\partial C(\Theta)(I-A(\Theta))^{-1}B(\Theta)}{\partial \Theta_i}\Big) + S\Big(\frac{\partial D(\Theta)}{\partial \Theta_i}\Big) \end{bmatrix} \tag{4.9}$$

Note that $\frac{\partial C(\Theta)(I-A(\Theta))^{-1}B(\Theta)}{\partial \Theta_i}$ from (4.9) can be expressed as

$$\begin{aligned} \frac{\partial C(\Theta)(I-A(\Theta))^{-1}B(\Theta)}{\partial \Theta_i} =& \frac{\partial C(\Theta)}{\partial \Theta_i}(I-A(\Theta))^{-1}B(\Theta) + \\ & C(\Theta)\frac{\partial (I-A(\Theta))^{-1}}{\partial \Theta_i}B(\Theta) + \\ & C(\Theta)(I-A(\Theta))^{-1}\frac{\partial B(\Theta)}{\partial \Theta_i}, \end{aligned} \tag{4.10}$$

and $\frac{\partial (I-A(\Theta))^{-1}}{\partial \Theta_i}$ from (4.10) is equal to

$$\frac{\partial (I-A(\Theta))^{-1}}{\partial \Theta_i} = (I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial \Theta_i}(I-A(\Theta))^{-1}. \tag{4.11}$$

Furthermore, note that $\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)^t}{\partial \Theta_i}$ $\forall t \in \{1, \ldots, j_n^*\}$ from (4.8) can be expressed as

$$\frac{\partial \bar{C}(\Theta)\bar{A}(\Theta)^t}{\partial \Theta_i} = \frac{\partial \bar{C}(\Theta)}{\partial \Theta_i}\bar{A}(\Theta)^t + \bar{C}(\Theta)\sum_{k=1}^{t}\left[\bar{A}(\Theta)^{k-1}\frac{\partial \bar{A}(\Theta)}{\partial \Theta_i}\bar{A}(\Theta)^{t-k}\right] \tag{4.12}$$

Considering (4.8), (4.9), (4.10), (4.11), (4.12), and noting that

$$\frac{\partial \bar{A}(\Theta)}{\partial \Theta_i} = \begin{bmatrix} \frac{\partial A(\Theta)}{\partial \Theta_i} & \frac{\partial B(\Theta)}{\partial \Theta_i} \\ 0 & 0 \end{bmatrix}, \quad \frac{\partial \bar{C}(\Theta)}{\partial \Theta_i} = \begin{bmatrix} \frac{\partial C(\Theta)}{\partial \Theta_i} & \frac{\partial D(\Theta)}{\partial \Theta_i} \end{bmatrix}, \tag{4.13}$$

it is clear that we have expressed $\frac{\partial H(\Theta)}{\partial \Theta_i}$ with all partial derivatives reduced to $\frac{\partial A(\Theta)}{\partial \Theta_i}$, $\frac{\partial B(\Theta)}{\partial \Theta_i}$, $\frac{\partial C(\Theta)}{\partial \Theta_i}$, and $\frac{\partial D(\Theta)}{\partial \Theta_i}$. Therefore, all that is needed to calculate $\frac{\partial H(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, in addition to the nominal system matrices, are $\frac{\partial A(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, $\frac{\partial B(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, $\frac{\partial C(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, and $\frac{\partial D(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$. An algorithm to efficiently compute $\frac{\partial H(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$ for all $i \in \{1, \ldots, q\}$ is presented in Algorithm 4.

Note that if analytical expressions for $\frac{\partial A(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, $\frac{\partial B(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, $\frac{\partial C(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, and $\frac{\partial D(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$ are unknown, then Algorithm 4 cannot be used to calculate $\frac{\partial H(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$. Thus we propose a simple numerical approximation of $\frac{\partial H(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}$, denoted by $\frac{\Delta H(\Theta)}{\Delta \Theta_i}\Big|_{\Theta_n}$, as follows:

$$\frac{\Delta H(\Theta)}{\Delta \Theta_i}\Big|_{\Theta_n} = \frac{H_n^{\Delta\Theta_i} - H_n}{\Delta \Theta_i} \tag{4.14}$$

where $\Delta\Theta_i$ is a small number and

$$H_n^{\Delta\Theta_i} = H(\ldots, \Theta_{n,i-1}, \Theta_{n,i} + \Delta\Theta_i, \Theta_{n,i+1}, \ldots). \tag{4.15}$$

The resulting approximation of $H_T(\Theta)$ is denoted $\hat{H}_T(\Theta)$:

$$\hat{H}_T(\Theta) = H_n + \sum_{i=1}^{q} \left.\frac{\Delta H(\Theta)}{\Delta \Theta_i}\right|_{\Theta_n} (\Theta_i - \Theta_{n,i}) \tag{4.16}$$

Note that $H_n^{\Delta\Theta_i}$ must be built up to $(j_n^* + 2)b$ rows to ensure that that $H_n^{\Delta\Theta_i} - H_n$ is a valid operation. Furthermore, it comes without surprise that as $\Delta\Theta_i$ approach 0, $\left.\frac{\Delta H(\Theta)}{\Delta\Theta_i}\right|_{\Theta_n}$ approaches $\left.\frac{\partial H(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}$. This behavior is demonstrated later in Figure 4.2.

Algorithm 4 $\left.\frac{\partial H(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}$

Inputs:
$A_n, \left.\frac{\partial A(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}, B_n, \left.\frac{\partial B(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}, C_n, \left.\frac{\partial C(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}, D_n, \left.\frac{\partial D(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}, S, j_n^*, \Theta_n$
Outputs:
$\left.\frac{\partial H(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}$
Notation:

- A_n, B_n, C_n, D_n are the respective $A(\Theta_n), B(\Theta_n), C(\Theta_n), D(\Theta_n)$ matrices
- I is the identity matrix of appropriate dimension

$\bar{A}_n := \begin{bmatrix} A_n & B_n \\ 0 & I \end{bmatrix}$
$\bar{C}_n := \begin{bmatrix} C_n & D_n \end{bmatrix}$
initialize i as 1
for each element of Θ_n **do**
  $\left.\frac{\partial \bar{A}(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} := \begin{bmatrix} \left.\frac{\partial A(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} & \left.\frac{\partial B(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} \\ 0 & 0 \end{bmatrix}$
  $\left.\frac{\partial \bar{C}(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} := \begin{bmatrix} \left.\frac{\partial C(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} & \left.\frac{\partial D(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} \end{bmatrix}$
  $X := \left.\frac{\partial C(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} (I - A_n)^{-1} B_n + C_n (I - A_n)^{-1} \left.\frac{\partial A(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} (I - A_n)^{-1} B_n + C_n (I - A_n)^{-1} \left.\frac{\partial B(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} + \left.\frac{\partial D(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}$
  initialize W as $[0,\ SX]$
  vertically concatenate $S \left.\frac{\partial \bar{C}(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}$ to the bottom of W, this forms the new W matrix
  initialize j as 1
  initialize P_1 as $\bar{A}_n^{-1} \left.\frac{\partial \bar{A}(\Theta)}{\partial\Theta_i}\right|_{\Theta_n}$
  initialize P_2 as 0
  while $j \leq j_n^*$ **do**
    $P_1 := \bar{A}_n P_1$
    $P_2 := P_2 \bar{A}_n + P_1$
    vertically concatenate $S \left(\left.\frac{\partial \bar{C}(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} \bar{A}_n^j + \bar{C}_n P_2 \right)$ to the bottom of W, this forms the new W matrix
    $j := j + 1$
  end while
  $\left.\frac{\partial H(\Theta)}{\partial\Theta_i}\right|_{\Theta_n} := W$
  $i := i + 1$
end for

We now discuss conditions for $\widetilde{O}_{\infty,T}(\Theta) = \widetilde{O}_{\infty}(\Theta)$. To begin, note that in general, $\widetilde{O}_{\infty,T}(\Theta)$ is an approximation of $\widetilde{O}_{\infty}(\Theta)$ because $H_T(\Theta)$ is a linearized version of $H(\Theta)$. Thus, determining when $H_T(\Theta) = H(\Theta)$ (i.e. when the linear representation is exact) determines when $O_{\infty,T}(\Theta) = O_{\infty}(\Theta)$. It follows that if $\frac{\partial^2 H(\Theta)}{\partial\Theta_i\partial\Theta_k} = 0,\ \forall i,k \in \{1,\ldots,q\}$, then $H_T(\Theta) = H(\Theta)$ for all Θ, which implies that $O_{\infty,T}(\Theta) = O_{\infty}(\Theta)$. A sufficient condition for $O_{\infty,T}(\Theta) = O_{\infty}(\Theta)$ is presented in the following theorem:

Theorem 3. *If* $\frac{\partial^2 A(\Theta)}{\partial\Theta_i\partial\Theta_k} = 0$, $\frac{\partial^2 B(\Theta)}{\partial\Theta_i\partial\Theta_k} = 0$, $\frac{\partial^2 C(\Theta)}{\partial\Theta_i\partial\Theta_k} = 0$, $\frac{\partial^2 D(\Theta)}{\partial\Theta_i\partial\Theta_k} = 0$, $\frac{\partial A(\Theta)}{\partial\Theta_i} = 0$, *and one of either* $\frac{\partial B(\Theta)}{\partial\Theta_i} = 0$ *or* $\frac{\partial C(\Theta)}{\partial\Theta_i} = 0$, *for all* $i,k \in [1,\ldots,q]$, *where* q *is the dimension of* Θ, *then* $O_{\infty,T}(\Theta) = O_{\infty}(\Theta)$.

The proof of Theorem 3 can be found in the Appendix. Note that we only specify that $O_{\infty,T}(\Theta)$ and $O_{\infty}(\Theta)$ are equivalent because the admissibility indices of $\widetilde{O}_{\infty,T}(\Theta)$ and $\widetilde{O}_{\infty}(\Theta)$, respectfully denoted by $j^*_{\Theta_n}$ and j^*_{Θ}, may not be equal. We discuss the conditions for $j^*_{\Theta_n} = j^*_{\Theta}$ next. The results are contained in Theorem 4.

Recall from Section 2.1.1, that the admissibility index is a function of the A, B, C, D matrices and the constraints. Therefore, it comes without surprise that the admissibility indices of two different systems with the same constraints may not be equal. In our case, we notice that the admissibility index of $\widetilde{O}_{\infty}(\Theta)$ for $\Theta \neq \Theta_n$, denoted j^*_{Θ}, is not necessarily equal to the admissibility index of $\widetilde{O}_{\infty}(\Theta_n)$, denoted $j^*_{\Theta_n}$. However, we wish to prove that for Θ in a small neighborhood of Θ_n, $j^*_{\Theta} = j^*_{\Theta_n}$. There are additional mild assumptions that we state later in Theorem 4 but we simplify the presentation here just to motivate what is to follow.

In preparation for Theorem 4, we review the algorithm provided in [22] to check for redundancy when building MAS. This algorithm is referred to in the proof of Theorem 4. Suppose $\widetilde{O}_{\infty}(\Theta)$ has been partially constructed with the inequalities in

(2.4) and (2.5) from $t = 0$ up to $t = j$, for some j. Let $H(\Theta)$ and h represent the matrices of this partially constructed MAS. We wish to test whether an inequality in (2.4) with $t = j + 1$ is redundant. To test for redundancy of a new row, we use the following linear program (LP) [22]:

$$f^*(\Theta, t, i) = \max \; c(\Theta, t, i)^T \bar{x} \quad \text{subject to} \quad H(\Theta, t)\bar{x} \leq h(t) \tag{4.17}$$

where in order to indicate the dependency of H and h on t, i.e., the prediction horizon corresponding to the number of rows of the partially constructed $\widetilde{O}_\infty(\Theta)$, we have modified the notation to $H(\Theta, t)$ and $h(t)$. Additionally, to show the dependency of the maximum cost, f^*, on Θ, t, and the constraint, we denote f^* as a function of Θ, t, and i, where i denotes the i^{th} element of s and $c(\Theta, t, i)^T \bar{x} \leq s_i$ corresponds to the row of $O_\infty(\Theta)$ being tested for redundancy. If $f^*(\Theta, t, i) \leq s_i$, the new inequality is redundant. Depending on the dimension of s, there may be multiple inequalities that need to be tested for each t. When this is the case, the admissibility index is reached if at time $t = j^*_\Theta + 1$, all b inequalities are deemed redundant.

In the following theorem, we wish to show, under mild assumptions stated in the theorem, that if Θ remains in a sufficiently small neighborhood of Θ_n, the admissibility index of $\widetilde{O}_\infty(\Theta)$ is the same as that for $\widetilde{O}_\infty(\Theta_n)$.

Theorem 4. *If $j^*_{\Theta_n}$ is found such that $f^*(\Theta_n, j^*_{\Theta_n} + 1, i) < s_i, \; \forall i \in \{1, 2, \ldots, b\}$, then $\exists \; \delta \in (0, R)$ such that $\|\Theta - \Theta_n\| < \delta$ implies that $j^*_\Theta = j^*_{\Theta_n}$.*

Proof. Because we assume $j^*_{\Theta_n}$ is found such that $f^*(\Theta_n, j^*_{\Theta_n} + 1, i) < s_i, \; \forall i \in$

$\{1, 2, \ldots, b\}$, there exists a real scalar $\zeta_1 > 0$ such that the following holds

$$\zeta_1 < s_i - f^*(\Theta_n, j^*_{\Theta_n} + 1, i), \ \forall i \in \{1, 2, \ldots, b\}. \tag{4.18}$$

Additionally, from the definition of j^*_Θ, we know that at $t = j^*_{\Theta_n}$, at least one inequality is not redundant. Therefore, we may conclude that there exists another real scalar $\zeta_2 > 0$ such that:

$$\exists i \in \{1, 2, \ldots, b\} \text{ such that } \ -\zeta_2 > s_i - f_i(\Theta_n, j^*_{\Theta_n}, i). \tag{4.19}$$

Finally, we know that we can find $\zeta = \min(\zeta_1, \zeta_2)$ that satisfies both (4.18) and (4.19).

Recall that the pair $(A(\Theta), C(\Theta))$ is observable for all $\Theta \in \mathbb{T}$, and that $\mathbb{Y}$ is compact, convex, and contains the origin. It follows from [22], that under these assumptions, $\widetilde{O}_\infty(\Theta)$ is also compact, convex, and contains the origin. This implies that for $\Theta = \Theta_n$, $t \in \{j^*_{\Theta_n}, j^*_{\Theta_n} + 1\}$, and $i \in \{1, 2, \ldots, b\}$, (4.17) results in a global maximum cost and, more specifically, the optimal solution is feasible and bounded. Furthermore, feasibility and boundedness of the primal LP implies the same for the dual [68]. Considering the boundedness of the maxima of both the primal and dual linear programs, it follows from Theorem 1.1 of [69] that the maximum cost of the primal LP, is continuous at Θ_n. Note that the maximum cost is not continuous in t or i because they are discrete variables. Continuity with respect to Θ at Θ_n implies that

$$\begin{aligned} &\forall \alpha > 0, \ \exists \delta > 0 \text{ such that} \\ &\forall t \in \{j^*_{\Theta_n}, j^*_{\Theta_n} + 1\}, \ \forall i \in \{1, 2, \ldots, b\}, \ \forall \Theta \in \mathbb{T}, \ \|\Theta - \Theta_n\| < \delta \implies \\ &\left| f^*(\Theta, t, i) - f^*(\Theta_n, t, i) \right| < \alpha. \end{aligned} \tag{4.20}$$

Note that δ does not depend on t or i. Note also that, since Θ in (4.20) satisfies $\Theta \in \mathbb{T}$, we may assume $\delta < R$ without loss of generality.

For the specific cases of $t = j^*_{\Theta_n} + 1$ and $t = j^*_{\Theta_n}$, and choosing $\alpha < \zeta$, we note the following two statements are true for all $i \in \{1, 2, \ldots, b\}$ and all Θ satisfying the conditions in (4.20):

$$f^*(\Theta, j^*_{\Theta_n} + 1, i) - f^*(\Theta_n, j^*_{\Theta_n} + 1, i) < \alpha < \zeta \tag{4.21}$$

$$f^*(\Theta, j^*_{\Theta_n}, i) - f^*(\Theta_n, j^*_{\Theta_n}, i) > -\alpha > -\zeta \tag{4.22}$$

Combining (4.18) and (4.21), we obtain the following result: $f^*(\Theta, j^*_{\Theta_n} + 1, i) - f^*(\Theta_n, j^*_{\Theta_n} + 1, i) < \alpha < \zeta \leq \zeta_1 < s_i - f^*(\Theta_n, j^*_{\Theta_n} + 1, i),\ \forall i \in \{1, 2, \ldots, b\}$, which implies that

$$f^*(\Theta, j^*_{\Theta_n} + 1, i) < s_i,\ \forall i \in \{1, 2, \ldots, b\}. \tag{4.23}$$

Similarly, Combining (4.19) and (4.22), we can conclude that, $\exists i \in \{1, 2, \ldots, b\}$ such that $f^*(\Theta, j^*_{\Theta_n}, i) - f^*(\Theta_n, j^*_{\Theta_n}, i) > -\alpha > -\zeta \geq -\zeta_2 > s_i - f^*(\Theta_n, j^*_{\Theta_n}, i)$. Therefore,

$$\exists i \in \{1, 2, \ldots, b\} \text{ such that } f^*(\Theta, j^*_{\Theta_n}, i) > s_i. \tag{4.24}$$

From (4.23) and (4.24), we can conclude that the admissibility index of $\widetilde{O}_\infty(\Theta)$ is $j^*_{\Theta_n}$, $\forall \Theta$ satisfying $\|\Theta - \Theta_n\| < \delta < R$. This implies that $j^*_\Theta = j^*_{\Theta_n}$. □

Note that Theorem 4 assumes that $f^*(\Theta_n, j^*_{\Theta_n} + 1, i) < s_i$ for all $i \in \{1, 2, \ldots, b\}$. However, mathematically, it is also possible that $f^*(\Theta_n, j^*_{\Theta_n} + 1, i) = s_i$ for some i (since the definition of redundancy requires $f^*(\Theta_n, j^*_{\Theta_n} + 1, i) \leq s_i$). If $\exists i \in \{1, 2, \ldots, b\}$ such that $f^*(\Theta_n, j^*_{\Theta_n} + 1, i) = s_i$, then (4.18) would become $0 \leq s_i -$

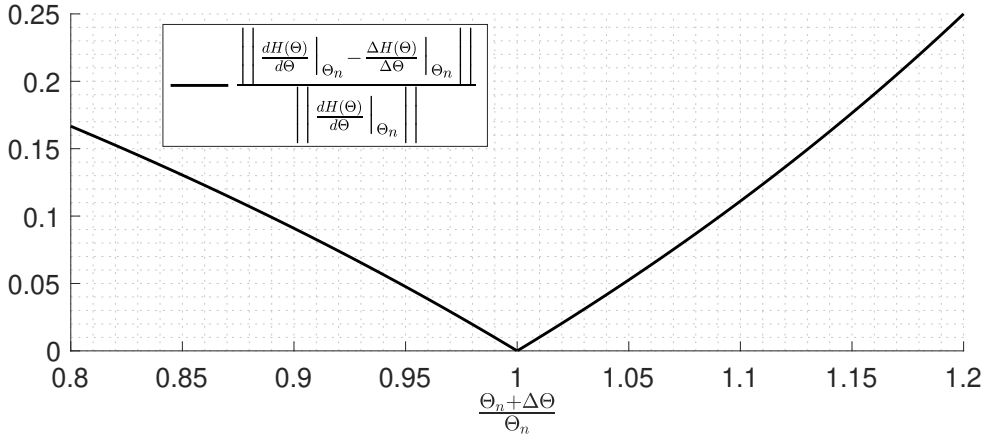

Figure 4.1: Comparison of $\frac{\Delta H(\Theta)}{\Delta\Theta}\Big|_{\Theta_n}$ *and* $\frac{dH(\Theta)}{d\Theta}\Big|_{\Theta_n}$ *for system* (4.25) *as a function of* $\Delta\Theta$*. Note that* $||\cdot||$ *represents the L2 norm.*

$f^*(\Theta_n, j^*_{\Theta_n}+1, i)$, and (4.21) would clearly be infeasible because $\zeta = 0 > \alpha > 0$ is logically false (i.e. no matter how small $\|\Theta-\Theta_n\| > 0$, $j^*_\Theta \neq j^*_{\Theta_n}$ and therefore the results of theorem do not hold). However, the set of all $\{\Theta_n, t, i\}$ for which $f^*(\Theta_n, t, i) - s_i = 0$ has measure 0, so this condition generically does not hold. Note that the previous analysis, along with Theorem 4, answers question 1 raised in Section 1.2.2.

We now illustrate some basic properties of Parameter-Dependent MAS via an example using the following system:

$$\begin{aligned} x(t+1) &= \Theta x(t) + 0.5v(t) \\ y(t) &= x(t) \\ \Theta_n &= 0.5 \end{aligned} \tag{4.25}$$

Figure 4.1 shows the effect of $\Delta\Theta$ on a measure of the difference between $\frac{\Delta H(\Theta)}{\Delta\Theta}\Big|_{\Theta_n}$ and $\frac{dH(\Theta)}{d\Theta}\Big|_{\Theta_n}$ for system (4.25). Note that as $\Delta\Theta$ approaches 0, $\frac{\Delta H(\Theta)}{\Delta\Theta}\Big|_{\Theta_n}$ approaches $\frac{dH(\Theta)}{d\Theta}\Big|_{\Theta_n}$.

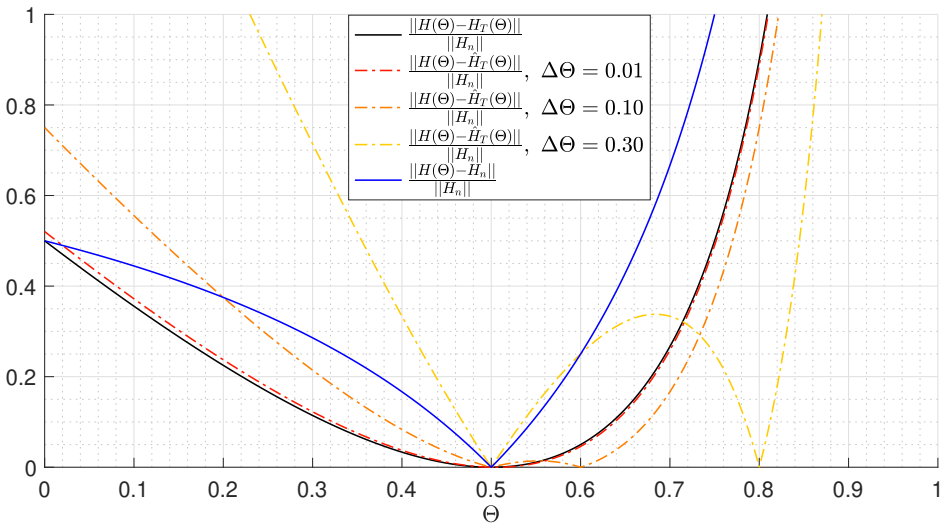

Figure 4.2: Comparison of $H_T(\Theta)$, H_n, and different $\hat{H}_T(\Theta)$ to $H(\Theta)$ at different values of Θ for system (4.25). Note that $||\cdot||$ represents the L2 norm.

In Figure 4.2, we compare $H_T(\Theta)$, H_n, and different $\hat{H}_T(\Theta)$ to $H(\Theta)$ at different values of Θ for system (4.25). Note that as Θ departs from Θ_n, the measure of difference of $H_T(\Theta)$ and H_n compared to $H(\Theta)$ increases as can be seen respectively in the black and blue traces of Figure 4.2. However, Note that $\frac{||H(\Theta)-H_T(\Theta)||}{||H_n||} \leq \frac{||H(\Theta)-H_n||}{||H_n||}$ for all $\Theta \in (0,1)$. Finally, note that as Θ departs from Θ_n, the measure of difference of the various $\hat{H}_T(\Theta)$ and $H(\Theta)$ increases locally around Θ_n but approaches 0 at $\Theta = \Theta_n + \Delta\Theta$. This phenomena occurs because the numerical sensitivity method calculates $\hat{H}_T(\Theta)$ such that $\hat{H}_T(\Theta) = H(\Theta)$ at $\Theta = \Theta_n$ and at $\Theta = \Theta_n + \Delta\Theta$. Note that while the Taylor method characterizes the change in MAS at a point, namely Θ_n, the numerical approximation of the Taylor method characterizes the change in MAS between two points, namely Θ_n and $\Theta_n + \Delta\Theta$. Therefore it follows that as $\Delta\Theta$ approaches 0, $\hat{H}_T(\Theta)$ approaches $H_T(\Theta)$. Note that this relationship is also portrayed in Figure 4.2.

Conditions for $\widetilde{O}_{\infty,T}(\Theta) = \widetilde{O}_{\infty}(\Theta)$ have been discussed; however, in general, the two sets will not be equivalent. Ultimately, it should be noted that for practical purposes, small deviations in the edges of $\widetilde{O}_{\infty,T}(\Theta)$ compared to those of $\widetilde{O}_{\infty}(\Theta)$ may become insignificant in the presence of external disturbances, measurement noise, and plant-model mismatch, which are all common in practical applications. Furthermore, note that techniques such as Pontryagin subtraction [15] can be used to account for uncertainty in the error of MAS approximation such that $\widetilde{O}_{\infty,T}(\Theta) \subset \widetilde{O}_{\infty}(\Theta), \forall t \geq 0$. Implementation of Parameter Varying MAS will be discussed later in Section 4.3.

4.2 λ-contractive MAS

For ease of reference, we review λ-contractive MAS for time-varying constraints as proposed in [23]. λ-contractive MAS is a constraint admissible set which decays quickly enough to guarantee enforcement of constraints that vary slowly with respect to time. The term "slowly" meaning that the constraints vary at a rate sufficiently slower than the dynamics of the closed loop system, which will be quantified later. To begin our review of λ-contractive MAS for time-varying constraints, consider the asymptotically stable system with decaying control input dynamics $v(t+1) = \lambda v(t)$, where $\lambda \in (0, 1)$:

$$\begin{aligned} x(t+1) &= Ax(t) + Bv(t) \\ y(t) &= Cx(t) \in \mathbb{Y}(t) \end{aligned} \tag{4.26}$$

where $x(t) \in \mathbb{R}^n$, $v(t) \in \mathbb{R}$, $y(t) \in \mathbb{R}^p$ and

$$\mathbb{Y}(t) := \{y : Sy(t) \leq s(t)\},\ s(t) > 0,\ 0 \in \mathbb{Y}(t)\ \forall t \geq 0 \tag{4.27}$$

is compact. By augmenting x and v we express (4.26) as follows:

$$\begin{aligned} \bar{x}(t+1) &= \bar{A}\bar{x}(t) \\ y(t) &= \bar{C}\bar{x}(t) \in \mathbb{Y}(t) \end{aligned} \tag{4.28}$$

where

$$\bar{x}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} A & B \\ 0 & \lambda \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix}$$

Note that (4.28) is asymptotically stable. In addition, we impose the following assumption on the constraint set regarding its maximum rate of contraction:

Assumption 1.

$$\lambda^* \mathbb{Y}(t) \subset \mathbb{Y}(t+1), \ \forall t \geq 0, \ \lambda^* > \lambda \tag{4.29}$$

We now define λ-contractive MAS for system (4.28). Recall that the goal is to maintain $y(t) \in \mathbb{Y}(t), \ \forall t \in \mathbb{Z}^+$, which, assuming that $y(0) \in \mathbb{Y}(0)$, can equivalently be expressed as:

$$y(t+1) \in \mathbb{Y}(t+1), \ \forall t \in \mathbb{Z}^+. \tag{4.30}$$

Based on Assumption 1, (4.30) is satisfied if:

$$y(t+1) \in \lambda \mathbb{Y}(t) \subset \lambda^* \mathbb{Y}(t) \subset \mathbb{Y}(t+1), \ \forall t \in \mathbb{Z}^+ \tag{4.31}$$

Therefore, to enforce (4.30), we define λ-contractive MAS, denoted $O_\infty^\lambda(t)$ as:

$$O_\infty^\lambda(t) \triangleq \left\{ \bar{x} : \ y(t+1) \in \lambda \mathbb{Y}(t), \ \forall t \in \mathbb{Z}^+ \right\} \tag{4.32}$$

which guarantees constraint satisfaction because, based on Assumption 1 and Theo-

rem 2.1 iv) of [22]:

$$\{\bar{x} : y(t+1) \in \lambda\mathbb{Y}(t)\} \subset \{\bar{x} : y(t+1) \in \mathbb{Y}(t+1)\} \tag{4.33}$$

Note that if $\lambda^* \geq 1$, there is no need for λ-contractive MAS because $\mathbb{Y}(t) \subseteq \mathbb{Y}(t+1)$ for all $t \geq 0$.

The formulation of the matrices that characterize $O_\infty^\lambda(t)$ is now presented. We begin by representing $y(t+1)$ from (4.32) in terms of $\bar{x}$:

$$y(t+1) \in \lambda\mathbb{Y}(t) \longrightarrow \bar{C}\bar{A}\bar{x}(t) \in \lambda\mathbb{Y}(t) \longrightarrow \bar{C}\frac{1}{\lambda}\bar{A}\bar{x}(t) \in \mathbb{Y}(t) \tag{4.34}$$

$O_\infty^\lambda(t)$ can equivalently be expressed as

$$O_\infty^\lambda(t) = \left\{\bar{x} : \ \bar{C}\frac{1}{\lambda}\bar{A}\bar{x}(t) \in \mathbb{Y}(t), \ \forall t \in \mathbb{Z}^+\right\}. \tag{4.35}$$

Therefore, the dynamics used when forming $O_\infty^\lambda(t)$ are:

$$\begin{aligned} \bar{x}_\lambda(t+1) &= \frac{1}{\lambda}\bar{A}\bar{x}_\lambda(t) \\ y_\lambda(t) &= \bar{C}\bar{x}_\lambda(t) \in \mathbb{Y}(t). \end{aligned} \tag{4.36}$$

Note that (4.36) is unstable if $\lambda \leq |\rho(A)|$, where $\rho(A)$ is the spectral radius of A. Thus, it is necessary that, $|\rho(A)| < \lambda < \lambda^*$.

From Section 2.1.1, we know that we can make $O_\infty^\lambda(t)$ finitely determined by contracting the steady state output of (4.36) to the interior of the constraint set:

$$y_\lambda(\infty) \triangleq C(I - \frac{1}{\lambda}A)^{-1}\frac{1}{\lambda}Bv \in (1-\epsilon)\mathbb{Y}(t) \tag{4.37}$$

Thus we represent the finitely determined version of $O_\infty^\lambda(t)$ as $\widetilde{O}_\infty^\lambda(t)$:

$$\widetilde{O}_\infty^\lambda(t) = \left\{\bar{x}_\lambda : H^\lambda \bar{x}_\lambda \leq h(t)\right\} \tag{4.38}$$

where

$$H^\lambda = \begin{bmatrix} 0 & SC(I - \frac{1}{\lambda}A)^{-1}\frac{1}{\lambda}B \\ & S\bar{C} \\ & S\bar{C}\frac{1}{\lambda}\bar{A} \\ & S\bar{C}\left(\frac{1}{\lambda}\bar{A}\right)^2 \\ & \vdots \\ & S\bar{C}\left(\frac{1}{\lambda}\bar{A}\right)^{j_\lambda^*} \end{bmatrix}, \qquad h(t) = \begin{bmatrix} (1-\epsilon)s(t) \\ s(t) \\ s(t) \\ s(t) \\ \vdots \\ s(t) \end{bmatrix} \tag{4.39}$$

and j_λ^* is the admissibility index of $\widetilde{O}_\infty^\lambda(t)$. Finally, note that j_λ^* is independent of positive scaling on $s(t)$ as explained in Lemma 1 of Section 3.1.1.

In Figure 4.3, we illustrate the difference between O_∞ and O_∞^λ for system (4.25). Note that in general, $O_\infty^\lambda \subset O_\infty$.

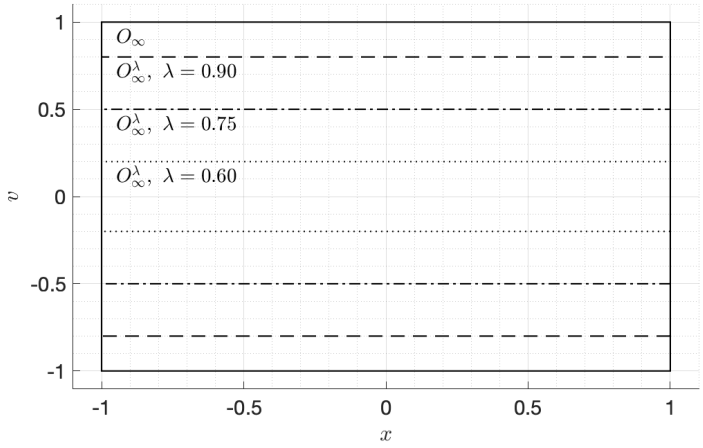

Figure 4.3: Comparison of O_∞ and O_∞^λ

4.3 Adaptive-Contractive Reference Governor (RG-AC)

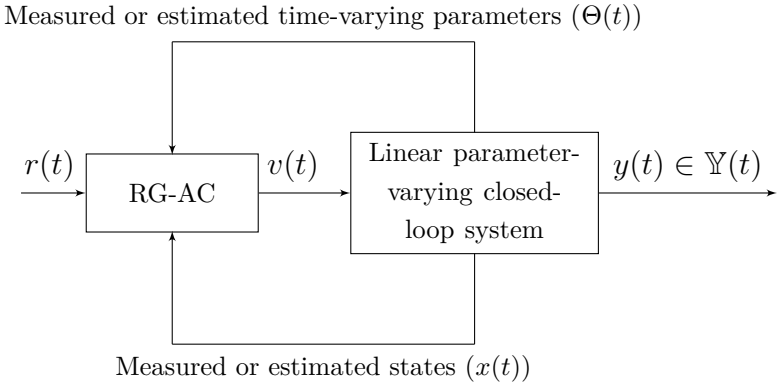

Figure 4.4: Block diagram of RG-AC governed system

As stated previously in this chapter, RG-AC employs a dynamic characterization of MAS using the linearization method discussed in Section 4.1 and the λ-contractive formulation introduced in Section 4.2. In this section, we generalize the characterization of MAS for RG-AC, namely $\widetilde{O}^{\lambda}_{\infty,T}(t)$, which is a λ-contractive version of $\widetilde{O}_{\infty,T}(\Theta)$ where Θ is now assumed to vary with time; which we indicate with the notation $\Theta(t)$. We also discuss the RG-AC algorithm as well as the computational aspects of RG-AC. Additionally, we present an illustrative example of RG-AC implemented in simulation.

Consider the system:

$$\begin{aligned} x(t+1) &= A(\Theta(t))x(t) + B(\Theta(t))v(t) \\ y(t) &= C(\Theta(t))x(t) \end{aligned} \tag{4.40}$$

subject to the convex polyhedral constraints:

$$y(t) \in \mathbb{Y}(t) := \{y : Sy(t) \leq s(t)\} \tag{4.41}$$

where $\Theta(t) \in \mathbb{R}^q$ is a time-varying parameter on which the elements of A, B, C and D depend on. Additionally, $s(t) \in \mathbb{R}^b,\ s(t) > 0$ for all $t \geq 0$; and $\mathbb{Y}(t)$ is nonempty, compact, and contains the origin for all $t \geq 0$. We denote the nominal system parameters as Θ_n, and assume there exists a closed neighborhood of Θ_n denoted by the set $\mathbb{T} = \{\Theta : \|\Theta - \Theta_n\| \leq R\}$ where $R \in \mathbb{R}^+$ and $\Theta(t) \in \mathbb{T}$ for all $t \geq 0$ such that the following assumption holds.

Assumption 2. *The system* (4.40) *is bounded-input bounded-state stable for all realizations of* $\Theta(t) \in \mathbb{T}$.

The formulation of $\widetilde{O}^{\lambda}_{\infty,T}(t)$ is similar to that of $\widetilde{O}_{\infty,T}(\Theta)$ with the additional requirement that $\widetilde{O}^{\lambda}_{\infty,T}(t)$ must be λ-contractive, and Θ is now assumed to vary slowly with time. Therefore, to build $\widetilde{O}^{\lambda}_{\infty,T}(t)$, refer to the formulation of $\widetilde{O}_{\infty,T}(\Theta)$ from Section 4.1 with the following exceptions: require that $v \in \mathbb{R}$, require that $D(\Theta) = 0$, replace $\mathbb{Y}$ from (4.2) with $\mathbb{Y}(t)$ from (4.41), replace $A(\Theta)$ and $B(\Theta)$ with $\frac{1}{\lambda}A(\Theta)$ and $\frac{1}{\lambda}B(\Theta)$ respectively.

To illustrate how $\widetilde{O}^{\lambda}_{\infty,T}(t)$ is updated every timestep, we first introduce notation for the matrices that describe $\widetilde{O}^{\lambda}_{\infty,T}(t)$:

$$\widetilde{O}^{\lambda}_{\infty,T}(t) = \{\bar{x} : H^{\lambda}_T(t)\bar{x} \leq h(t)\} \tag{4.42}$$

Recall that

$$H^{\lambda}_T(t) = H^{\lambda}_n + H^{\lambda}_s(t) \tag{4.43}$$

where H_n^λ is the λ contractive version of H_n, and $H_s^\lambda(t)$ is the λ contractive, parameter varying version of $H_s(\Theta)$ (refer to (4.4)):

$$H_s^\lambda(t) = \sum_{i=1}^{q} \frac{\partial H^\lambda(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n} (\Theta_i(t) - \Theta_{n,i}). \tag{4.44}$$

Note that because $\Theta(t)$ is now assumed to vary with time, we feedback the measured or estimated values of $\Theta(t)$ into (4.44) to update $\widetilde{O}_{\infty,T}^\lambda(t)$ as $\Theta(t)$ varies. This feedback loop is portrayed in Figure 4.4.

Now that $\widetilde{O}_{\infty,T}^\lambda(t)$ has been established for RG-AC, we introduce the algorithm for computing $v(t)$ at every timestep. Recall that in our formulation of $\widetilde{O}_{\infty,T}^\lambda(t)$, we imposed contractive dynamics on v, namely $v(t+1) = \lambda v(t)$. Therefore, in RC-AG update law, $v(t)$ is calculated by the convex combination of $\lambda v(t-1)$ and $r(t)$ so that $\kappa(t) = 0$ is feasible for all realizations of $\Theta(t)$ and $\mathbb{Y}(t)$:

$$v(t) = \lambda v(t-1) + \kappa \left(r(t) - \lambda v(t-1)\right) \tag{4.45}$$

Note that when $\kappa = 0$, the update law becomes $v(t) = \lambda v(t-1)$ which is consistent with the dynamics used to create $\widetilde{O}_{\infty,T}^\lambda(t)$ and allows $v(t)$ to approach zero during a threat of constraint violation. Finally, note that (4.45) also allows RG-AC to recover from constraint violations because the contractive modified reference allows for $v(t)$ to approach zero during constraint violation regardless of $r(t)$, which in turn attracts the states of the asymptotically stable system to the origin which is guaranteed to belong to $\widetilde{O}_{\infty,T}^\lambda(t)$ (see (4.41)). However, if λ is designed properly by considering the maximum radial contraction ratio of $\widetilde{O}_{\infty,T}^\lambda(t)$ based on the rate limits of $\Theta(t)$ and $\mathbb{Y}(t)$, theoretical constraint satisfaction can be guaranteed. This answers question 2

from section 1.2.2.

Now that we have established $\widetilde{O}^{\lambda}_{\infty,T}(t)$ and the update law for RG-AC, the optimization problem for κ at every timestep follows similarly to Section 2.1.1, where we aim to minimize the distance between $v(t)$ and $r(t)$ while maintaining $(x(t), v(t)) \in \widetilde{O}^{\lambda}_{\infty,T}(t)$. The RG-AC optimization problem is thus:

$$\begin{aligned} \underset{\kappa\in[0,1]}{\text{maximize}} \quad & \kappa \\ \text{s.t.} \quad & v(t) = \lambda v(t-1) + \kappa\left(r(t) - \lambda v(t-1)\right) \\ & (x(t),\ v(t)) \in \widetilde{O}^{\lambda}_{\infty,T}(t) \end{aligned} \tag{4.46}$$

where $x(t)$ are the measured or observed states of the pre-stabilized closed-loop system (see Figure 4.4). For completeness, the RG-AC algorithm is presented in Algorithm 5, where it is assumed that $\Theta(t)$ is an input. Note that the only significant difference in the algorithms for RG (Algorithm 2) and RG-AC (Algorithm 5) is the need to update the H_x and H_v matrices in RG-AC based on $\Theta(t)$. Thus the computational demand of RG-AC is comparable to that of RG if $\Theta(t)$ is measured. However, if $\Theta(t)$ cannot be measured, the introduction of a parameter estimation algorithm will increase the computational complexity of RG-AC to a level significantly greater than that of RG. This completes the answer to question 3 raised in Section 1.2.2. The answer to question 4 is presented next in Theorem 5, where we discuss the stability of the RG-AC loop.

Theorem 5. *The RG-AC loop is BIBO stable.*

Proof. From (4.45), and with $\kappa \in [0, 1]$, it follows that $v(t)$ is a convex combination of $r(t)$ and $\lambda v(t-1)$, where $\lambda \in (0, 1)$, both of which are bounded (the bounds are

0 and $r(t)$). Therefore, $v(t)$ is bounded. Boundedness of $v(t)$ and Assumption 2 on system (4.40) imply BIBO stability of the governed system. □

Note that this result is similar to the stability result of the standard RG with the exception that $v(t)$ does not necessarily converge to $r(t)$ let alone any constant due to the nature of the time-varying system and constraints. However, note that $v(t)$ is bounded between 0 and $r(t)$.

Algorithm 5 RG-AC

Inputs:

$r(t), x(t), v(t-1), H_n^\lambda, \frac{\partial H^\lambda(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n}, h(t), \Theta_n, \Theta(t), \lambda$

Output:

$v(t)$

initialize H_s^λ as 0
initialize κ as a vector with the same number of rows as h
initialize i as 1
initialize j as 1
for each element of Θ_n **do**
    $H_s^\lambda := \frac{\partial H^\lambda(\Theta)}{\partial \Theta_i}\Big|_{\Theta_n} \big(\Theta(t)(\text{row } i) - \Theta_n(\text{row } i)\big) + H_s^\lambda$
    $i := i + 1$
end for
$H_T^\lambda := H_n^\lambda + H_s^\lambda$
assign H_x as all but the last column of H_T^λ
assign H_v as the last column of H_T^λ
for each row in $\widetilde{0}_{\infty,T}^\lambda$ **do**
    $n := h(t)(\text{row } j) - H_x(\text{row } j)x(t) - H_v(\text{row } j)v(t-1)$
    $d := H_v(\text{row } j)(r(t) - v(t-1))$
    $\kappa(\text{row } j) := \text{kappa}(n, d)$
    $j := j + 1$
end for
$\kappa^* := \min(\kappa)$
$v(t) := \lambda v(t-1) + \kappa^*(r(t) - \lambda v(t-1))$

function kappa(n, d)

if $n > 0$ **then**
    if $d > 0$ **then**
        $\kappa := \min(n/d, 1)$
    else
        $\kappa := 1$
    end if
else
    $\kappa := 0$
end if
return κ

end function

4.3.1 Illustrative Examples

Example 1: time-varying parameter in A matrix

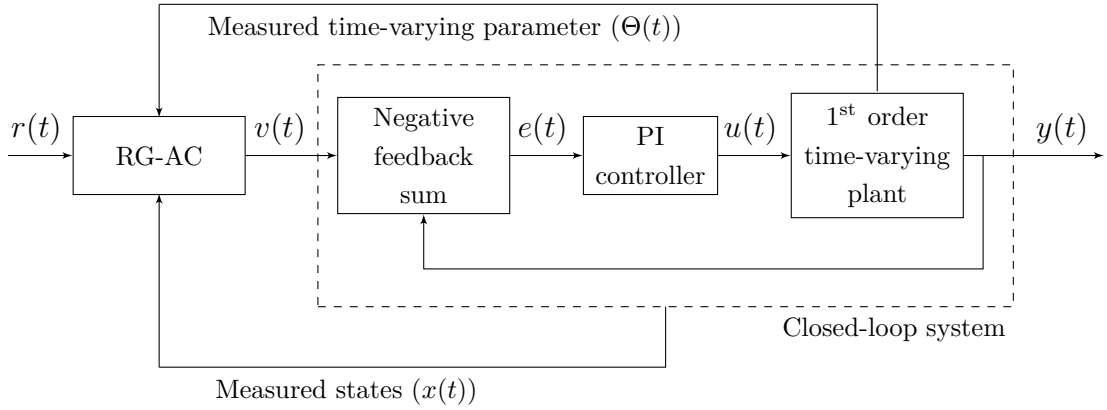

Figure 4.5: Block diagram of the RG-AC governed system used for simulation

To illustrate the efficacy of RG-AC, we show simulation results of RG-AC governing a linear parameter-varying (LPV) system subject to a time-varying constraint on the system's output. To begin with this example, we first introduce the parameter-varying closed loop system, which consists of a LPV plant, and a PI controller. The LPV plant is described by the following discrete-time state-space model:

$$\begin{aligned} z(t+1) &= A_p(\Theta)z(t) + B_p u(t) \\ y(t) &= C_p z(t) \end{aligned} \tag{4.47}$$

where $A_p(\Theta) = \Theta(t) = \Theta_n + 0.1\sin(0.01t)$, $\Theta_n = 0.3$, $B_p = 0.1$, and $C_p = 1$.

Additionally, the PI controller is of the form:

$$\begin{aligned} w(t+1) &= w(t) + e(t) \\ u(t) &= Ke(t) + K_i w(t) \end{aligned} \tag{4.48}$$

where $w(t)$ is the integrator state, $K = 1$ is the proportional gain, and $K_i = 1.5$ is the integral gain. Noting that $e(t) = v(t) - y(t)$, the entire closed-loop system model takes on the form:

$$\begin{aligned} x(t+1) &= A(\Theta)x(t) + Bu(t) \\ y(t) &= Cx(t) \\ x = \begin{bmatrix} z \\ w \end{bmatrix}, \quad A(\Theta) &= \begin{bmatrix} \Theta(t) - B_p C_p K & B_p K_i \\ -C_p & 1 \end{bmatrix}, \\ B = \begin{bmatrix} B_p K \\ 1 \end{bmatrix}, \quad C &= \begin{bmatrix} C_p & 0 \end{bmatrix} \end{aligned} \tag{4.49}$$

In this example, the constraint is chosen to be $|y(t)| \leq s_i(t) = 1 + A_0 \sin(\omega_0 t)$, where subscript i denotes the i^{th} element of $s(t)$, $A_0 = 0.3$, and $\omega_0 = 0.02$. The constraint set evolves according to $\mathbb{Y}(t+1) = \frac{s_i(t+1)}{s_i(t)}\mathbb{Y}(t)$ for all $t \in \mathbb{Z}^+$. The minimum of $\frac{s_i(t+1)}{s_i(t)}$ occurs when $\omega_0(t + \frac{1}{2}) = \pi$ and thus $t = \frac{\pi}{\omega_0} - \frac{1}{2}$. Therefore, for this example

$$\lambda^* = \frac{s_i(\pi/\omega_0 + 1/2)}{s_i(\pi/\omega_0 - 1/2)} = 0.994 \tag{4.50}$$

Additionally, note that $\lambda \leq |\rho(A(\Theta_n)| = 0.7$, which means that $0.7 < \lambda < 0.994$. However, due to the possibility of additional contractions of $\widetilde{O}^{\lambda}_{\infty,T}(t)$ caused by $\Theta(t)$, we shrink the upper bound of λ by a small amount. In final, for this example,

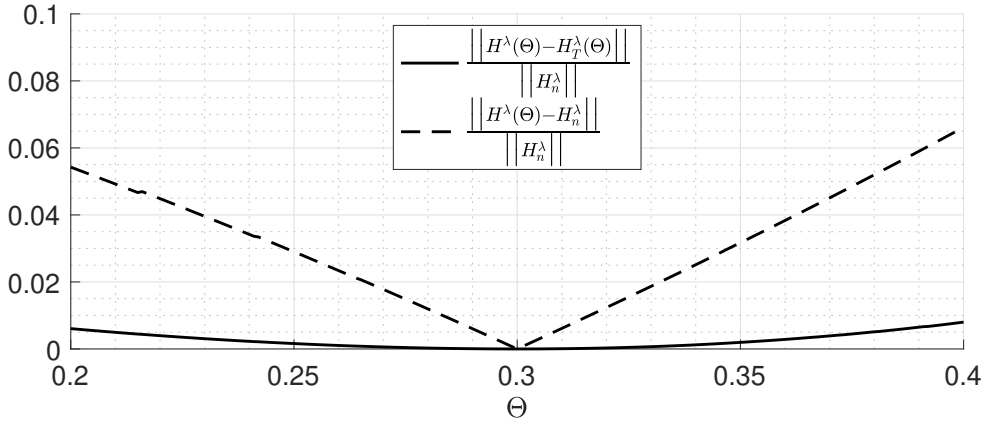

Figure 4.6: Comparison of $H_T^\lambda(\Theta)$ and H_n^λ to $H^\lambda(\Theta)$ for different values of Θ. Note that $||\cdot||$ represents the L2 norm.

$\lambda = 0.99$.

With the contractive dynamics on v applied to system (4.49), we obtain the dynamics used to create H_n^λ as follows:

$$\begin{aligned} \bar{x}_\lambda(t+1) &= \frac{1}{\lambda}\bar{A}(\Theta_n)\bar{x}_\lambda(t) \\ y_\lambda(t) &= \bar{C}\bar{x}_\lambda(t) \\ \bar{x}_\lambda = \begin{bmatrix} x_\lambda \\ v_\lambda \end{bmatrix}, \quad \bar{A}(\Theta_n) &= \begin{bmatrix} A(\Theta_n) & B \\ 0 & \lambda \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} C & 0 \end{bmatrix} \end{aligned} \tag{4.51}$$

Note that in this example, Θ is a scalar. Thus (4.44) consists of only one term, namely $H_s^\lambda(t) = \frac{dH(\Theta)}{d\Theta}\Big|_{\Theta_n}(\Theta(t) - \Theta_n)$, which can be computed using 4.8, 4.10, 4.11, 4.12, and 4.13; where

$$\frac{dA(\Theta)}{d\Theta} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad \frac{dB}{d\Theta} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \frac{dC}{d\Theta} = \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad \frac{dD}{d\Theta} = 0 \tag{4.52}$$

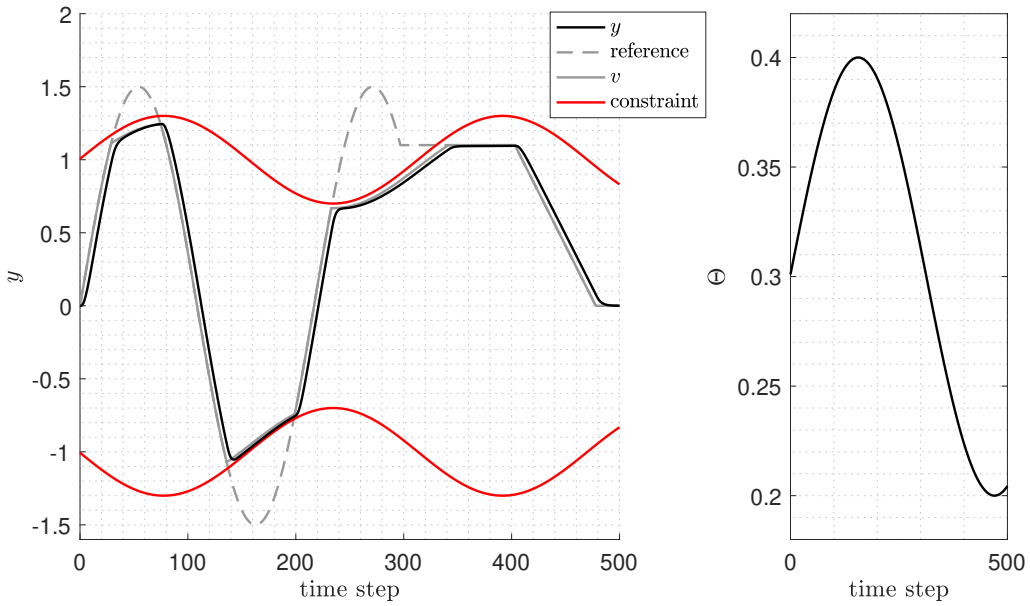

Figure 4.7: RG-AC simulation

Figure 4.6 compares $H_T^\lambda(\Theta)$ and H_n^λ to $H^\lambda(\Theta)$ for all $\Theta \in [0.2, 0.4]$, which is the range of $\Theta(t)$ for this example. It is clear from the figure that the linearization method ($H_T^\lambda(\Theta)$) greatly improves the approximation of $H^\lambda(\Theta)$ compared to no adaptive method (H_n^λ).

The simulation results are shown in Figure 4.7. Note that constraints are satisfied for all time steps and that RG-AC is able to pull v towards the origin and away from the reference when the constraint tightens (see time steps 140 through 200). In Figure 4.8, we compare RG-AC to MPC, for the same simulation conditions. For this example, MPC was implemented to control and enforce constraints on the entire closed loop system. Note from Figure 4.8, that MPC is able to over-drive the reference to the closed loop system, thus increasing tracking performance. Furthermore note that MPC is generally able to drive the output closer to the constraints compared to

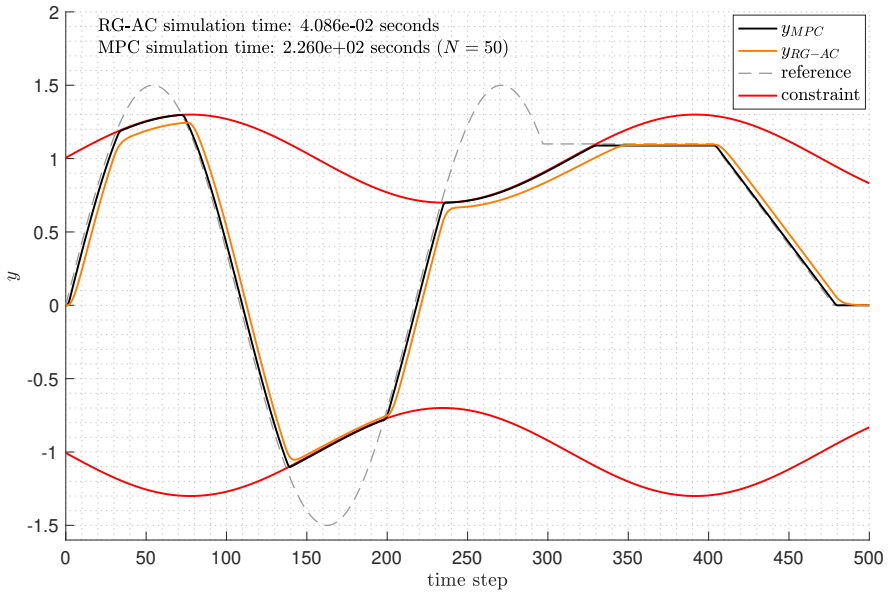

Figure 4.8: RG-AC vs. MPC simulation

RG-AC. The drawback of MPC however is computation time. From Figure 4.8, it is evident that the simulation time of MPC was close to 4 orders of magnitude greater than the simulation time of RG-AC. We compare RG-AC to Command Governor (CG) [37, 38, 70] in Figure 4.9. Note that, unlike MPC, CG does not over-drive the reference. However, compared to RG-AC, CG is able to drive the output closer to the constraint. Like MPC, the drawback of CG lies within the computational complexity of the online optimization problems. Similar to MPC, the simulation time of CG was close to 4 orders of magnitude greater than the simulation time of RG-AC. The reason that MPC and CG both have longer computation times compared to AC-RG is because they solve a quadratic program at every timestep whereas RG-AC solves an explicit linear program at every timestep.

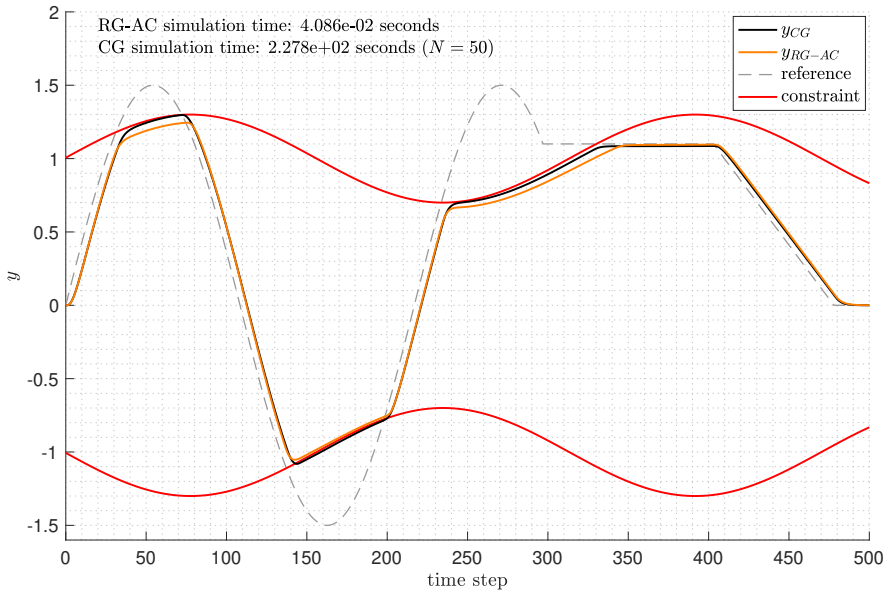

Figure 4.9: RG-AC vs. CG simulation

Example 2: time-varying parameter in B matrix

Consider Figure 4.4 in which RG-AC governs the "closed-loop system" described by the following dynamics:

$$\begin{aligned} x(t+1) &= Ax(t) + B(\Theta(t))u(t) \\ y(t) &= Cx(t) \end{aligned}$$
$$A = \begin{bmatrix} 0.2 & 0.15 \\ -1 & 1 \end{bmatrix}, \quad B(\Theta(t)) = \begin{bmatrix} 0.1\Theta(t) \\ 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{4.53}$$

where $\Theta(t) = \Theta_n + 5\sin(0.01t)$, $\Theta_n = 10$. Furthermore, suppose that the time-varying constraint is equivalent to that of Example 1.

Figure 4.10 compares $H_T^\lambda(\Theta)$ and H_n^λ to $H^\lambda(\Theta)$ for all $\Theta \in [5, 15]$, which is the

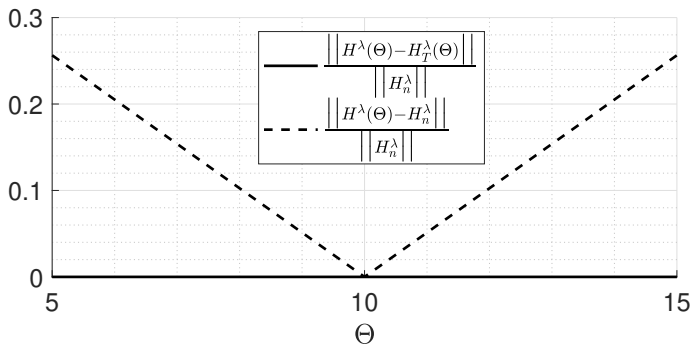

Figure 4.10: Comparison of $H_T^\lambda(\Theta)$ and H_n^λ to $H^\lambda(\Theta)$ for different values of Θ for Example 2. Note that $||\cdot||$ represents the L2 norm.

range of $\Theta(t)$ for this example. It is clear from the figure that the linearization method in this example is exact, which agrees with Theorem 3 because Θ only appears in the B matrix and $\frac{d^2B}{d\Theta^2} = 0$. Simulation results of RG-AC governing system (4.53) are shown in Figure 4.11. Note that constraints are satisfied for all time, and that the response is minimally conservative. If we remove the adaptive nature of the MAS and only use contractive MAS formed from the nominal realization of the system, then constraints are violated as shown in Figure 4.12. Furthermore, the system performance continues to degrade if we substitute RG-AC for RG which is evident in Figure 4.13. For reference to the systems ungoverned behavior, see Figure 4.14.

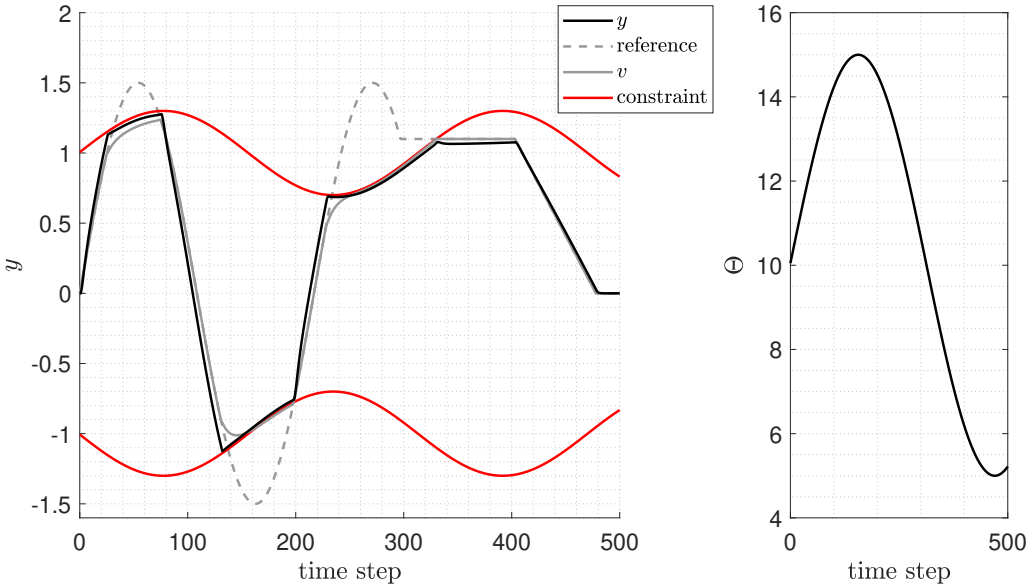

Figure 4.11: RG-AC simulation

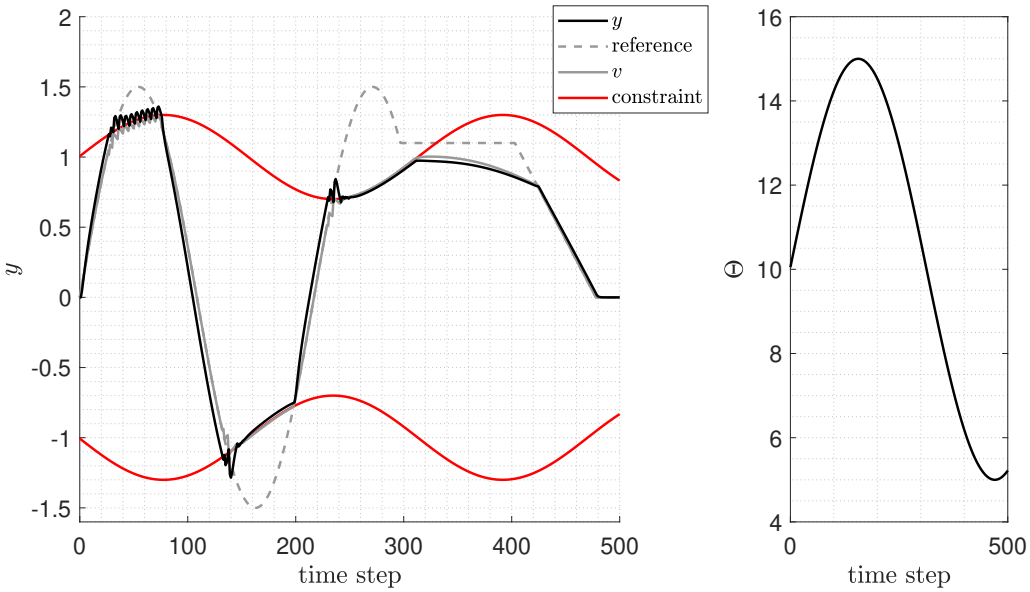

Figure 4.12: Contractive RG simulation

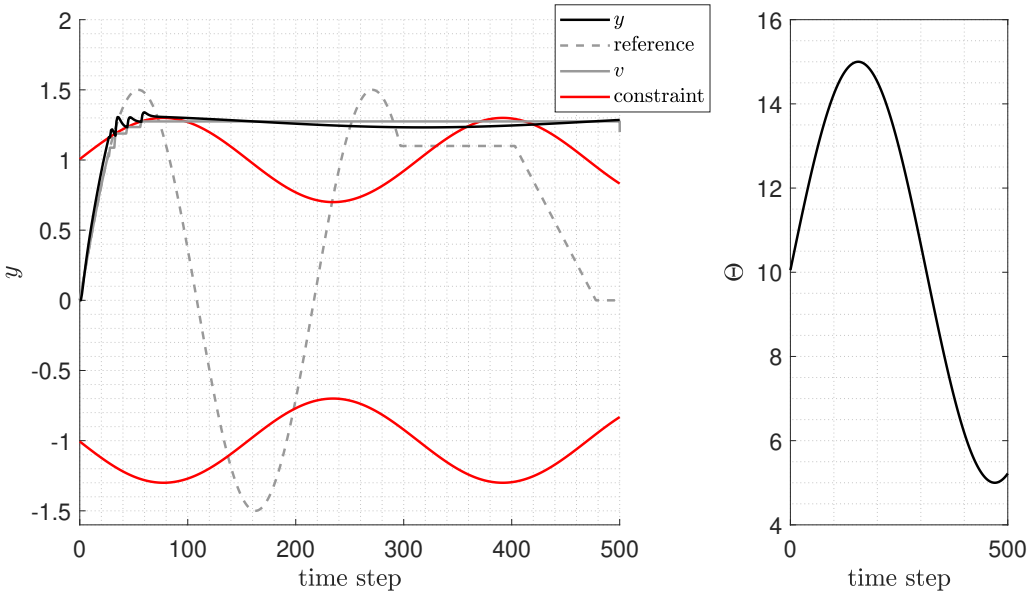

Figure 4.13: RG simulation

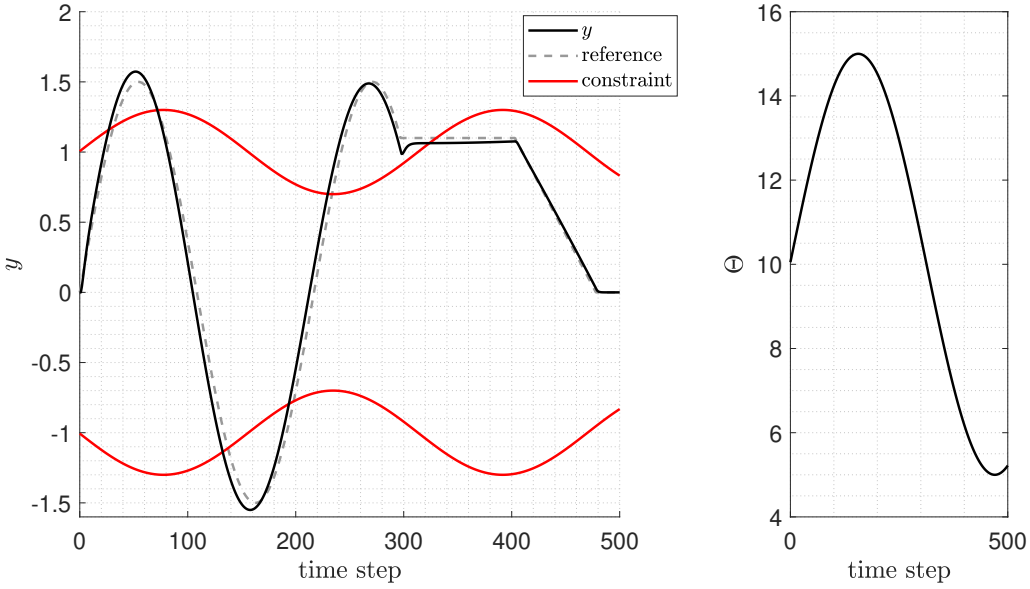

Figure 4.14: Ungoverned simulation

Chapter 5

Conclusions

5.1 Summary

In Chapter 3, an overshoot mitigation control scheme was developed using the reference governor framework. The solution, known as the Reference Governor with Dynamic Constraint (RG-DC), utilizes a dynamic maximal admissible set (MAS) to constrain the tracking output such that overshoot of step inputs is eliminated. The RG-DC loop was proven to be BIBO stable. Additionally, properties of the dynamic MAS were studied and theorems were proven that allow for RG-DC to operate without recalculation of the matrices that define the dynamic MAS. While RG-DC can guarantee overshoot elimination for all step inputs with the proper initial conditions, it may not remove overshoot for a more general time-varying reference r. Conditions were provided in the paper under which elimination will be guaranteed for time-varying r. Regarding real-world applications, RG-DC is useful for limiting overshoot of systems with legacy tracking controllers where overshoot is not desired. RG-DC allows the user to execute a simple algorithm before the input of the closed loop system instead of modifying a well established, pre-existing tracking controller to mitigate overshoot of the tracking output.

In Chapter 4, an efficient constraint management scheme based off the reference governor framework was developed to enforce time-varying constraints on parameter-varying systems. The solution known as Adaptive-Contractive Reference Governor (RG-AC) employs a time-varying characterization of MAS designed for fast updates between sampling instances as the parameters of the closed-loop system vary around their nominal values. The fast update times allow for RG-AC to operate on systems with fast processes and short sample times while the dynamic nature of Parameter-

Dependent MAS reduces the conservativeness of the constrained response compared to alternative methods that utilize robust positively invariant sets. It was explained that the dynamic characterization of MAS can be exact if the time-varying parameters appear in certain pats of the system. Additionally, the stability of the RG-AC loop was proven. RG-AC is suited for applications where parameter-varying systems require constraint enforcement but the constraints are not necessarily safety critical. The latter suggestion stems from the fact that the dynamic characterization of MAS is not necessarily an exact representation of MAS for the parameter-varying system. Note however that the dynamic characterization of MAS can be made robust to linearization error by means of Pontryagin subtraction (as was done in [32]) or correct choice of the contraction parameter λ.

5.2 Future Works

Future work on RG-DC and its effect on frequency response are of interest. More specifically, we would like to study the settling time of systems under RG-DC and explore the application of the RG-DC as a nonlinear filter.

Future work on RG-AC involves understanding the effect of parameter variation rate on the required contraction ratio of Adaptive MAS to ensure constraint admissibility.

Bibliography

[1] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918 – 927, 2009.

[2] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control lyapunov–barrier function. *Automatica*, 66:39 – 47, 2016.

[3] Keng Peng Tee and Shuzhi Sam Ge. Control of nonlinear systems with partial state constraints using a barrier lyapunov function. *International Journal of Control*, 84(12):2008–2023, 2011.

[4] Jan Marian Maciejowski. *Predictive control: with constraints*. Prentice Hall, 2008.

[5] Eduardo F Camacho and Carlos Bordons Alba. *Model predictive control*. Springer Science & Business Media, 2013.

[6] Graham Goodwin, María M Seron, and José A De Doná. *Constrained control and estimation: an optimisation approach*. Springer Science & Business Media, 2006.

[7] Wook Hyun Kwon and Soo Hee Han. *Receding horizon control: model predictive control for state models*. Springer Science & Business Media, 2006.

[8] Manfred Morari and Jay H Lee. Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23(4-5):667–682, 1999.

[9] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.

[10] Y. Wang and S. Boyd. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, 18(2):267–278, 2010.

[11] I. Kolmanovsky, E. Garone, and S. Di Cairano. Reference and command governors: A tutorial on their theory and automotive applications. In *Proc. American Control Conference*, pages 226–241, June 2014.

[12] E. G. Gilbert and I. Kolmanovsky. Discrete-time reference governors for systems with state and control constraints and disturbance inputs. In *Proc. IEEE Conference on Decision and Control*, volume 2, pages 1189–1194, Dec 1995.

[13] Emanuele Garone, Stefano Di Cairano, and Ilya Kolmanovsky. Reference and command governors for systems with constraints: A survey on theory and applications. *Automatica*, 75:306–328, 2017.

[14] Elmer G. Gilbert and Ilya Kolmanovsky. Fast reference governors for systems with state and control constraints and disturbance inputs. *International Journal of Robust and Nonlinear Control*, 9(15):1117–1141, 1999.

[15] Ilya V. Kolmanovsky and E. G. Gilbert. Maximal output admissible sets for discrete-time systems with disturbance inputs. In *Proc. American Control Conference*, volume 3. IEEE, 1995.

[16] A. Bemporad. Reference governor for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 43(3):415–419, 1998.

[17] H. R. Ossareh. Reference governors and maximal output admissible sets for linear periodic systems. *International Journal of Control*, pages 1–13, 2019.

[18] J. Osorio and H. R. Ossareh. A stochastic approach to maximal output admissible sets and reference governors. In *Proc. IEEE Conference on Control Technology and Applications*, pages 704–709, Aug 2018.

[19] Aidan Laracy and Hamid Ossareh. Constraint management for batch processes using iterative learning control and reference governors. In *Proceedings of Machine Learning Research (PMLR)*, 2020.

[20] Joycer Osorio, Mario Santillo, Julia Buckland Seeds, Mrdjan Jankovic, and Hamid R Ossareh. A reference governor approach towards recovery from constraint violation. In *2019 American Control Conference (ACC)*, pages 1779–1785. IEEE, 2019.

[21] Yudan Liu, Joycer Osorio, et al. Decoupled reference governors for multi-input multi-output systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 1839–1846. IEEE, 2018.

[22] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, Sep. 1991.

[23] U. Kalabić and I. Kolmanovsky. Reference and command governors for systems with slowly time-varying references and time-dependent constraints. In *Proc. IEEE Conference on Decision and Control*, Dec 2014.

[24] S. Devasia. Should model-based inverse inputs be used as feedforward under plant uncertainty? *IEEE Transactions on Automatic Control*, 47(11):1865–1871, Nov 2002.

[25] Qingze Zou and Santosh Devasia. Precision preview-based stable-inversion for nonlinear nonminimum-phase systems: The VTOL example. *Automatica*, 43:117–127, 01 2007.

[26] Thanh Hung Tran, Quang Phuc Ha, and Hung T. Nguyen. Robust non-overshoot time responses using cascade sliding mode-PID control. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 11:1224–1231, 2007.

[27] Y. Tange, S. Kiryu, and T. Matsui. Overshoot suppression control based on final tracking error estimation and quantifier elimination. In *Proc. Asian Control Conference*, pages 1–6, May 2015.

[28] U. Kalabić, I. Kolmanovsky, J. Buckland, and E. Gilbert. Reference and extended command governors for control of turbocharged gasoline engines based on linear models. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 319–325, Sep. 2011.

[29] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor. The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. In *Proc. American Control Conference*, volume 2, pages 804–809, June 2005.

[30] Magnus Nilsson, Emil Klintberg, Philipp Rumschinski, and Lars Johannesson Mårdh. Admissible sets for slowly-varying discrete-time systems. *Automatica*, 112:108676, 2020.

[31] A. Chakrabarty, K. Berntorp, and S. Di Cairano. Learning-based parameter-adaptive reference governors. In *2020 American Control Conference (ACC)*, pages 956–961, 2020.

[32] Jing Sun and I. V. Kolmanovsky. Load governor for fuel cell oxygen starvation protection: a robust nonlinear reference governor approach. *IEEE Transactions on Control Systems Technology*, 13(6):911–920, 2005.

[33] Petros Kapasouris, Michael Athans, and Gunter Stein. Design of feedback control systems for unstable plants with saturating actuators. *National Aeronautics and Space Administration, Washington, DC*, 1988.

[34] E. G. Gilbert, I. Kolmanovsky, and Kok Tin Tan. Nonlinear control of discrete-time linear systems with state and control constraints: a reference governor with global convergence properties. In *Proceedings of 1994 33rd IEEE Conference on Decision and Control*, volume 1, pages 144–149 vol.1, 1994.

[35] Alberto Bemporad and Edoardo Mosca. Nonlinear predictive reference filtering for constrained tracking. In *Proc. European Control Conf.*, pages 1720–1725, 1995.

[36] Elmer G Gilbert, Ilya Kolmanovsky, and Kok Tin Tan. Discrete-time reference governors and the nonlinear control of systems with state and control constraints. *International Journal of robust and nonlinear control*, 5(5):487–504, 1995.

[37] A. Bemporad, A. Casavola, and E. Mosca. Nonlinear control of constrained linear systems via predictive reference management. *IEEE Transactions on Automatic Control*, 42(3):340–349, 1997.

[38] A. Casavola, E. Mosca, and D. Angeli. Robust command governors for constrained linear systems. *IEEE Transactions on Automatic Control*, 45(11):2071–2077, 2000.

[39] Uros Kalabić. *Reference governors: Theoretical Extensions and Practical Applications.* PhD thesis, University of Michigan, Ann Arbor, 2015.

[40] U. V. Kalabić, J. H. Buckland, S. L. Cooper, S. K. Wait, and I. V. Kolmanovsky. Reference governors for enforcing compressor surge constraints. *IEEE Transactions on Control Systems Technology*, 24(5):1729–1739, 2016.

[41] H. Nakada, P. Martin, G. Milton, A. Iemura, and A. Ohata. An application study of online reference governor to boost pressure control for automotive diesel engines. In *2014 American Control Conference*, pages 3135–3140, 2014.

[42] Hayato Nakada, Gareth Milton, Peter Martin, Akiyuki Iemura, and Akira Ohata. Application of reference governor using soft constraints and steepest descent method to diesel engine aftertreatment temperature control. *SAE International Journal of Engines*, 6:257–266, 05 2013.

[43] Robert Gaynor, Fabian Mueller, Faryar Jabbari, and Jacob Brouwer. On control concepts to prevent fuel starvation in solid oxide fuel cells. *Journal of Power Sources*, 180(1):330 – 342, 2008.

[44] Salah Laghrouche, Imad Matraji, Fayez Shakil Ahmed, Samir Jemei, and Maxime Wack. Load governor based on constrained extremum seeking for pem fuel cell oxygen starvation and compressor surge protection. *International Journal of Hydrogen Energy*, 38(33):14314 – 14322, 2013.

[45] A. Vahidi, I. Kolmanovsky, and A. Stefanopoulou. Constraint handling in a fuel cell system: A fast reference governor approach. *IEEE Transactions on Control Systems Technology*, 15(1):86–98, 2007.

[46] S. J. Moura, N. A. Chaturvedi, and M. Krstić. Constraint management in li-ion batteries: A modified reference governor approach. In *2013 American Control Conference*, pages 5332–5337, 2013.

[47] K. A. Smith. Electrochemical control of lithium-ion batteries [applications of control]. *IEEE Control Systems Magazine*, 30(2):18–25, 2010.

[48] I. V. Kolmanovsky, E. G. Gilbert, and H. E. Tseng. Constrained control of vehicle steering. In *2009 IEEE Control Applications, (CCA) Intelligent Control, (ISIC)*, pages 576–581, 2009.

[49] Meir Pachter and Russel B Miller. Manual flight control with saturating actuators. *IEEE Control Systems Magazine*, 18(1):10–20, 1998.

[50] Russel B Miller. Manual tracking flight control with amplitude and rate constrained dynamic actuators. Technical report, Air Force Institute of Technology, 1997.

[51] Domenico Famularo, Davide Martino, and Massimiliano Mattei. Constrained control strategies to improve safety and comfort on aircraft. *Journal of guidance, control, and dynamics*, 31(6):1782–1792, 2008.

[52] D. Martino. Flight control with amplitude and rate constraints: A command governor approach. In *2008 American Control Conference*, pages 1788–1793, 2008.

[53] W. Lucia, M. Sznaier, and G. Franzè. An obstacle avoidance and motion planning command governor based scheme: The qball-x4 quadrotor case of study. In *53rd IEEE Conference on Decision and Control*, pages 6135–6140, 2014.

[54] Marco M. Nicotra, Mihovil Bartulovic, Emanuele Garone, and Bruno Sinopoli. A distributed explicit reference governor for constrained control of multiple uavs. *IFAC-PapersOnLine*, 48(22):156 – 161, 2015. 5th IFAC Workshop on Distributed Estimation and Control in Networked Systems NecSys 2015.

[55] Alessandro Casavola, Domenico Famularo, Giuseppe Franzé, and Emanuele Garone. Set-points reconfiguration in networked multi-area electrical power systems. *International journal of adaptive control and signal processing*, 23(8):808–832, 2009.

[56] Alessandro Casavola, Giuseppe Franze, Francesco Tedesco, and Emanuele Garone. Distributed coordination-by-constraint strategies in networked multi-area power systems. In *2011 IEEE International Symposium on Industrial Electronics*, pages 1697–1702. IEEE, 2011.

[57] Francesco Tedesco and Alessandro Casavola. Fault-tolerant distributed load/frequency supervisory strategies for networked multi-area microgrids. *International Journal of Robust and Nonlinear Control*, 24(8-9):1380–1402, 2014.

[58] Alessandro Casavola, Giuseppe Franzè, Daniele Menniti, and Nicola Sorrentino. Voltage regulation in distribution networks in the presence of distributed generation: A voltage set-point reconfiguration approach. *Electric power systems research*, 81(1):25–34, 2011.

[59] Francesco Tedesco and Alessandro Casavola. A distributed command governor approach for voltage regulation in medium voltage power grids with distributed generation. In *2013 American Control Conference*, pages 3492–3497. IEEE, 2013.

[60] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[61] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[62] Kyoung-jae Kim. Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1-2):307–319, 2003.

[63] Daniel Alspach and Harold Sorenson. Nonlinear bayesian estimation using gaussian sum approximations. *IEEE transactions on automatic control*, 17(4):439–448, 1972.

[64] T. Manrique, M. Fiacchini, T. Chambrion, and G. Millerioux. Mpc tracking under time-varying polytopic constraints for real-time applications. In *2014 European Control Conference (ECC)*, pages 1480–1485, 2014.

[65] Wen Li and Jason Meiners. Introduction to phase-locked loop system modeling. *Analog Applications*, 2000.

[66] A. Pavlov, N. van de Wouw, and H. Nijmeijer. Frequency response functions and bode plots for nonlinear convergent systems. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 3765–3770, 2006.

[67] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 1996.

[68] Robert J. Vanderbei. *Linear Programming Foundations and Extensions Fourth Edition*. Springer, 2014.

[69] D. Martin. On the continuity of the maximum in parametric linear programming. *Journal of Optimization Theory and Applications - J OPTIMIZ THEOR APPL*, 17:205–210, 05 1975.

[70] D. Angeli and E. Mosca. Command governors for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 44(4):816–820, 1999.

Chapter 6

Appendix

The proof of Theorem 3 can be found below:

In general, $\frac{\partial^2 H(\Theta)}{\partial \Theta_i \partial \Theta_k}$ can be expressed as follows:

$$\frac{\partial^2 H(\Theta)}{\partial \Theta_i \partial \Theta_k} = \begin{bmatrix} \lim_{j\to\infty} S\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)^j}{\partial \Theta_i \partial \Theta_k} \\ S\frac{\partial^2 \bar{C}(\Theta)}{\partial \Theta_i \partial \Theta_k} \\ S\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)}{\partial \Theta_i \partial \Theta_k} \\ S\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)^2}{\partial \Theta_i \partial \Theta_k} \\ \vdots \\ S\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)^{j^*_n}}{\partial \Theta_i \partial \Theta_k} \end{bmatrix} \tag{6.1}$$

where

$$\lim_{j\to\infty} S\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)^j}{\partial \Theta_i \partial \Theta_k} = \begin{bmatrix} 0 & S\left(\frac{\partial^2 C(\Theta)(I - A(\Theta))^{-1}B(\Theta)}{\partial \Theta_i \partial \Theta_k} + \frac{\partial^2 D(\Theta)}{\partial \Theta_i \partial \Theta_k}\right) \end{bmatrix} \tag{6.2}$$

Note that $\frac{\partial^2 C(\Theta)(I-A(\Theta))^{-1}B(\Theta)}{\partial\Theta_i\partial\Theta_k} + \frac{\partial^2 D(\Theta)}{\partial\Theta_i\partial\Theta_k}$ from (6.2) can be expressed as

$$\begin{aligned}
\frac{\partial^2 C(\Theta)(I-A(\Theta))^{-1}B(\Theta)}{\partial\Theta_i\partial\Theta_k} + \frac{\partial^2 D(\Theta)}{\partial\Theta_i\partial\Theta_k} = \\
\frac{\partial^2 C(\Theta)}{\partial\Theta_i\partial\Theta_k}(I-A(\Theta))^{-1}B(\Theta) \\
+\frac{\partial C(\Theta)}{\partial\Theta_k}(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_i}(I-A(\Theta))^{-1}B(\Theta) \\
+\frac{\partial C(\Theta)}{\partial\Theta_k}(I-A(\Theta))^{-1}\frac{\partial B(\Theta)}{\partial\Theta_i} \\
+\frac{\partial C(\Theta)}{\partial\Theta_i}(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_k}(I-A(\Theta))^{-1}B(\Theta) \\
+C(\Theta)(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_i}(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_k}(I-A(\Theta))^{-1}B(\Theta) \\
+C(\Theta)(I-A(\Theta))^{-1}\frac{\partial^2 A(\Theta)}{\partial\Theta_i\partial\Theta_k}(I-A(\Theta))^{-1}B(\Theta) \\
+C(\Theta)(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_k}(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_i}(I-A(\Theta))^{-1}B(\Theta) \\
+C(\Theta)(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_k}(I-A(\Theta))^{-1}\frac{\partial B(\Theta)}{\partial\Theta_i} \\
+\frac{\partial C(\Theta)}{\partial\Theta_i}(I-A(\Theta))^{-1}\frac{\partial B(\Theta)}{\partial\Theta_k} \\
+C(\Theta)(I-A(\Theta))^{-1}\frac{\partial A(\Theta)}{\partial\Theta_i}(I-A(\Theta))^{-1}\frac{\partial B(\Theta)}{\partial\Theta_k} \\
+C(\Theta)(I-A(\Theta))^{-1}\frac{\partial^2 B(\Theta)}{\partial\Theta_i\partial\Theta_k} \\
+\frac{\partial^2 D(\Theta)}{\partial\Theta_i\partial\Theta_k}
\end{aligned} \tag{6.3}$$

and $\frac{\partial^2 \bar{C}(\Theta)}{\partial\Theta_i\partial\Theta_k}$ from (6.2) from is equivalent to

$$\frac{\partial^2 \bar{C}(\Theta)}{\partial\Theta_i\partial\Theta_k} = \begin{bmatrix} \frac{\partial^2 C(\Theta)}{\partial\Theta_i\partial\Theta_k} & \frac{\partial^2 D(\Theta)}{\partial\Theta_i\partial\Theta_k} \end{bmatrix} \tag{6.4}$$

Furthermore, note that $\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)^t}{\partial\Theta_i\partial\Theta_k}$ $\forall t \in \{1,\ldots,j_n^*\}$ from (6.1) can be expressed as

$$
\begin{aligned}
\frac{\partial^2 \bar{C}(\Theta)\bar{A}(\Theta)^t}{\partial\Theta_i\partial\Theta_k} = \\
&\left[\begin{array}{cc} \frac{\partial^2 C(\Theta)}{\partial\Theta_i\partial\Theta_k} & \frac{\partial^2 D(\Theta)}{\partial\Theta_i\partial\Theta_k} \end{array}\right]\bar{A}(\Theta)^t \\
&+\left[\begin{array}{cc} \frac{\partial C(\Theta)}{\partial\Theta_k} & \frac{\partial D(\Theta)}{\partial\Theta_k} \end{array}\right]\sum_{l=1}^{t}\left[\bar{A}(\Theta)^{l-1}\left[\begin{array}{cc} \frac{\partial A(\Theta)}{\partial\Theta_i} & \frac{\partial B(\Theta)}{\partial\Theta_i} \\ 0 & 0 \end{array}\right]\bar{A}(\Theta)^{t-l}\right] \\
&+\left[\begin{array}{cc} \frac{\partial C(\Theta)}{\partial\Theta_i} & \frac{\partial D(\Theta)}{\partial\Theta_i} \end{array}\right]\sum_{l=1}^{t}\left[\bar{A}(\Theta)^{l-1}\left[\begin{array}{cc} \frac{\partial A(\Theta)}{\partial\Theta_k} & \frac{\partial B(\Theta)}{\partial\Theta_k} \\ 0 & 0 \end{array}\right]\bar{A}(\Theta)^{t-l}\right] \\
&+\bar{C}(\Theta)\sum_{l=1}^{t}\left[\sum_{g=1}^{l-1}\left[\bar{A}(\Theta)^{g-1}\left[\begin{array}{cc} \frac{\partial A(\Theta)}{\partial\Theta_i} & \frac{\partial B(\Theta)}{\partial\Theta_i} \\ 0 & 0 \end{array}\right]\bar{A}(\Theta)^{l-g-1}\right]\left[\begin{array}{cc} \frac{\partial A(\Theta)}{\partial\Theta_k} & \frac{\partial B(\Theta)}{\partial\Theta_k} \\ 0 & 0 \end{array}\right]\bar{A}(\Theta)^{t-l}\right] \\
&+\bar{C}(\Theta)\sum_{l=1}^{t}\left[\bar{A}(\Theta)^{l-1}\left[\begin{array}{cc} \frac{\partial^2 A(\Theta)}{\partial\Theta_i\partial\Theta_k} & \frac{\partial^2 B(\Theta)}{\partial\Theta_i\partial\Theta_k} \\ 0 & 0 \end{array}\right]\bar{A}(\Theta)^{t-l}\right] \\
&+\bar{C}(\Theta)\sum_{l=1}^{t}\left[\bar{A}(\Theta)^{l-1}\left[\begin{array}{cc} \frac{\partial A(\Theta)}{\partial\Theta_k} & \frac{\partial B(\Theta)}{\partial\Theta_k} \\ 0 & 0 \end{array}\right]\sum_{g=1}^{t-l}\left[\bar{A}(\Theta)^{g-1}\left[\begin{array}{cc} \frac{\partial A(\Theta)}{\partial\Theta_i} & \frac{\partial B(\Theta)}{\partial\Theta_i} \\ 0 & 0 \end{array}\right]\bar{A}(\Theta)^{t-l-g}\right]\right]
\end{aligned}
\tag{6.5}
$$

Note that if all terms of (6.3), (6.4), and (6.5) for all $t > 0$ equal 0, then $\frac{\partial^2 H(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$ and thus $O_{\infty,T}(\Theta) = O_\infty(\Theta)$.

We begin with the first term of the right-hand-side of (6.3), which requires that $\frac{\partial^2 C(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$ for the term to equal zero. Next, the second term of the right-hand-side of (6.3) requires that either $\frac{\partial C(\Theta)}{\partial \Theta_k} = 0$ or $\frac{\partial A(\Theta)}{\partial \Theta_i} = 0$. If we methodically continue this process through every term of (6.3), (6.4), and (6.5), the following sufficient condition is realized: if $\frac{\partial^2 A(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$, $\frac{\partial^2 B(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$, $\frac{\partial^2 C(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$, $\frac{\partial^2 D(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$, $\frac{\partial A(\Theta)}{\partial \Theta_i} = 0$, and one of either $\frac{\partial B(\Theta)}{\partial \Theta_i} = 0$ or $\frac{\partial C(\Theta)}{\partial \Theta_i} = 0$, for all $i, k \in [1, \ldots, q]$, where q is the dimension of Θ, then $\frac{\partial^2 H(\Theta)}{\partial \Theta_i \partial \Theta_k} = 0$, and thus $O_{\infty,T}(\Theta) = O_\infty(\Theta)$.